ОБЪЕДИНЕННЫЙ
ИНСТИТУТ
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ

Дубна

E11-96-106

G.A.Emel'yanenko, T.T.Rakhmonov, E.B.Dushanov

ALGORITHMS AND PROGRAMS OF THE CRITICAL-COMPONENT METHOD OF INVERSION OF TRIDIAGONAL MATRICES AND SOLUTION OF SYSTEMS OF LINEAR EQUATIONS

Submitted to «Математическое моделирование»

1996

1. Introduction

The purpose of the present paper is :
Development of new effective methods and algorithms of inversion of tridiagonal matrices of the general form and of solution of systems of linear algebraic equations with those matrices; creation of complexes of new programs on the basis of the developed algorithms.

2. Methods of inversion of tridiagonal matrices of the general form C and of solution of systems of linear algebraic equations $CX = Y$

In this sect., based on the representations of C and $B = C^{-1}$ found in refs. [$1 \div 3$], we develop an effective method for inversion of matrices C, that takes the structure of $B = C^{-1}$ into account. We also present the critical–component method worked out in [1] for solving the systems of linear algebraic equations $CX = Y$.

Let C be a nonsingular real tridiagonal matrix of the general form

$$C = \begin{bmatrix} q_1 & r_2 & & & \\ p_2 & q_2 & r_3 & & \\ & \ddots & \ddots & \ddots & \\ & & p_{m-1} & q_{m-1} & r_m \\ & & & p_m & q_m \end{bmatrix} \equiv \begin{bmatrix} \left[C_1^{l_1-2}\right]_{r_{l_1-1}} & & & & \\ p_{l_1-1}\begin{bmatrix} q_{l_1-1} & r_{l_1} \\ p_{l_1} & q_{l_1} \end{bmatrix}_{r_{l_1+1}} & & & & \\ & p_{l_1+1}\left[C_{l_1+1}^{l_2-2}\right]_{r_{l_2-1}} & & & \\ & & \cdots\cdots & & \\ & & & p_{l_n-1}\begin{bmatrix} q_{l_n-1} & r_{l_n} \\ p_{l_n} & q_{l_n} \end{bmatrix}_{r_{l_n+1}} & \\ & & & & p_{l_n+1}\left[C_{l_n+1}^{m}\right] \end{bmatrix}, \quad (2.1)$$

where

$$C_{l_k+1}^{l_{k+1}-2} = \begin{bmatrix} q_{l_k+1} & r_{l_k+2} & & \\ p_{l_k+2} & q_{l_k+2} & r_{l_k+3} & \\ \ddots & \ddots & \ddots & \\ & p_{l_{k+1}-3} & q_{l_{k+1}-3} & r_{l_{k+1}-2} \\ & & p_{l_{k+1}-2} & q_{l_{k+1}-2} \end{bmatrix}, k = 0, 1, ..., n, l_0 = 0, l_{n+1} - 2 = m, \quad (2.2)$$

$\{p_i \neq 0\}_{i=2}^{m}$ are subdiagonal elements, $\{r_i \neq 0\}_{i=2}^{m}$ are off–diagonal ones, $\{q_i\}_{i=1}^{m}$ are diagonal elements and $C_{l_k+1}^{l_{k+1}-2} = tridiag\{q_{l_k+1}, q_i, p_i, r_i\}_{i=l_k+2}^{l_{k+1}-2}$ are submatrices of the matrix $C(2.1)$.

Block decomposition of the type (2.1) $\div$ (2.2) of the initial matrix C is required for the construction [3] of effective direct numerical methods of solution of the above–mentioned algebraic problems.

Below, we'll report only those of the results from refs. $1 \div 3$ for which we will here develop practical algorithms and programs; namely, in refs. [1,2] we have obtained the following representations for matrices inverse of C, $B = C^{-1}$:

Representation 2.1 (of the type $B = \mathring{B} + B_1 Z$)

$$B = \underbrace{\begin{bmatrix}
[\tilde{B}_1^{l_1-2}] & & & & & & \\
& \begin{bmatrix}0 & \\ & 0\end{bmatrix} & & & & & \\
& & [\tilde{B}_{l_1+1}^{l_2-2}] & & & & \\
& & & \begin{bmatrix}0 & \\ & 0\end{bmatrix} & & & \\
& & & & [\tilde{B}_{l_2+1}^{l_3-2}] & & \\
& & & & & \ddots & \\
& & & & & \begin{bmatrix}0 & \\ & 0\end{bmatrix} & \\
& & & & & & [\tilde{B}_{l_n+1}^{m}]
\end{bmatrix}}_{\mathring{B}} +$$

$$+ \underbrace{\begin{bmatrix}
\begin{bmatrix}(\tilde{B}_{1l_1-2}r_{l_1-1}) & 0\\ \vdots & \vdots\\ (\tilde{B}_{l_1-2l_1-2}r_{l_1-1}) & 0\end{bmatrix} & & & \\
\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix} & & & \\
\begin{bmatrix}0 & (\tilde{B}_{l_1+1l_1+1}p_{l_1+1})\\ \vdots & \vdots\\ 0 & (\tilde{B}_{l_2-2l_1+1}p_{l_1+1})\end{bmatrix} & \begin{bmatrix}(\tilde{B}_{l_1+1l_2-2}r_{l_2-1}) & 0\\ \vdots & \vdots\\ (\tilde{B}_{l_2-2l_2-2}r_{l_2-1}) & 0\end{bmatrix} & & \\
& \begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix} & & \\
& \begin{bmatrix}0 & (\tilde{B}_{l_2+1l_2+1}p_{l_2+1})\\ \vdots & \vdots\\ 0 & (\tilde{B}_{l_3-2l_2+1}p_{l_2+1})\end{bmatrix} & \cdots & \begin{bmatrix}(\tilde{B}_{l_{n-1}+1l_n-2}r_{l_n-1}) & 0\\ \vdots & \vdots\\ (\tilde{B}_{l_n-2l_n-2}r_{l_n-1}) & 0\end{bmatrix}\\
& & & \begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}\\
& & & \begin{bmatrix}0 & (\tilde{B}_{l_n+1l_n+1}p_{l_n+1})\\ \vdots & \vdots\\ 0 & (\tilde{B}_{ml_n+1}p_{l_n+1})\end{bmatrix}
\end{bmatrix}}_{B_1} \cdot Z, \qquad (2.3)$$

where $\mathring{B}$ is a block–diagonal matrix of dimension $m \times m$, B_1 is a rectangular matrix of dimension $m \times 2n$, $\tilde{B}_{ij}$ are elements of the last (first) columns of

submatrices $\tilde{B}_\rho^\nu = [C_\rho^\nu]^{-1}$, where $[C_\rho^\nu]^{-1}$ is an inverse of C_ρ^ν. Here Z is a rectangular matrix of dimension $2n \times m$ being [1,2] a solution of the following system of matrix equations:

$$\underbrace{\begin{bmatrix}
[Q_{l_1}][R_{l_2}] & & \\
[P_{l_2}][Q_{l_2}][R_{l_3}] & & \\
\cdots\cdots & & \\
& & [P_{l_n}][Q_{l_n}]
\end{bmatrix}}_{\Omega}
\underbrace{\begin{bmatrix}
[Z_{l_1l_1-2}][Z_{l_1l_1}][Z_{l_1l_2-2}][Z_{l_1l_2}][Z_{l_1l_3-2}]\cdots[Z_{l_1l_n}][Z_{l_1m}]\\
[Z_{l_2l_1-2}][Z_{l_2l_1}][Z_{l_2l_2-2}][Z_{l_2l_2}][Z_{l_2l_3-2}]\cdots[Z_{l_2l_n}][Z_{l_2m}]\\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
[Z_{l_nl_1-2}][Z_{l_nl_1}][Z_{l_nl_2-2}][Z_{l_nl_2}][Z_{l_nl_3-2}]\cdots[Z_{l_nl_n}][Z_{l_nm}]
\end{bmatrix}}_{Z} =$$

$$\underbrace{\begin{bmatrix}
[b_{l_1l_1-2}]\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}[b_{l_1l_2-2}] & & \\
& [b_{l_2l_2-2}]\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}[b_{l_2l_3-2}] & \\
& \cdots\cdots\cdots\cdots & \\
& & [b_{l_nl_n-2}]\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}[b_{l_nm}]
\end{bmatrix}}_{B_2}, \qquad (2.4)$$

where Ω is a nonsingular [2] tridiagonal matrix of dimension $2n \times 2n$, B_2 is a rectangular matrix of dimension $2n \times m$,

$$\begin{aligned}
Q_{l_i} &= \begin{bmatrix}(q_{l_i-1} - p_{l_i-1}\tilde{B}_{l_i-2l_i-2}r_{l_i-1}) & (r_{l_i})\\ (p_{l_i}) & (q_{l_i} - r_{l_i+1}\tilde{B}_{l_i+1l_i+1}p_{l_i+1})\end{bmatrix}, \quad R_{l_i} = \begin{bmatrix}0 & 0\\ (-r_{l_{i-1}+1}\tilde{B}_{l_{i-1}+1l_i-2}r_{l_i-1}) & 0\end{bmatrix},\\
P_{l_i} &= \begin{bmatrix}0 & (-p_{l_i-1}\tilde{B}_{l_i-2l_{i-1}+1}p_{l_{i-1}+1})\\ 0 & 0\end{bmatrix},\\
b_{l_il_i-2} &= \begin{bmatrix}(p_{l_i-1}\tilde{B}_{l_i-2l_{i-1}+1}) \cdots (p_{l_i-1}\tilde{B}_{l_i-2l_i-2})\\ 0 \quad \cdots \quad 0\end{bmatrix},\\
b_{l_il_{i+1}-2} &= \begin{bmatrix}0 \quad \cdots \quad 0\\ (r_{l_i+1}\tilde{B}_{l_i+1l_i+1}) \cdots (r_{l_i+1}\tilde{B}_{l_i+1l_{i+1}-2})\end{bmatrix}, \quad i = 1, 2, \ldots n,
\end{aligned} \qquad (2.5)$$

where $\tilde{B}_{ij}$ are elements of the first (last) rows of submatrices $\tilde{B}_\rho^\nu = [C_\rho^\nu]^{-1}$,

$$Z_{l_il_j-2} = \begin{bmatrix}B_{l_i-1l_{j-1}+1} \cdots B_{l_i-1l_j-2}\\ B_{l_il_{j-1}+1} \cdots B_{l_il_j-2}\end{bmatrix}, \quad Z_{l_il_i} = \begin{bmatrix}B_{l_i-1l_i-1} & B_{l_i-1l_i}\\ B_{l_il_i-1} & B_{l_il_i}\end{bmatrix}. \qquad (2.6)$$

So, the determination of $B = C^{-1}$ is reduced to the solution of the system of equations $(2.4) \div (2.6)$. The matrix Ω of this system is ill - posed [2,3] in the general case if so does the initial matrix C.

In refs. [1,2], we have introduced the following generalized sequences*):

$$\begin{cases} \Lambda_{i+1} = q_i - p_i \Lambda_i^{-1} r_i, \ \Lambda_2 = q_1, \ i = 2, ..., m, \\ \text{if } \Lambda_i \neq 0 \text{ for all } 2 \leq i \leq m. \\ \text{If } \Lambda_i = 0 \text{ for any } i \text{ from } (2 \leq i \leq m), \text{ then} \\ \Lambda_{i+1} \text{ - undefined, but } \Lambda_{i+2} = q_{i+1}; \end{cases} \tag{2.6$'$}$$

$$\begin{cases} G_{i-1} = q_i - r_{i+1} G_i^{-1} p_{i+1}, \ G_{m-1} = q_m, \ i = m-1, ..., 1, \\ \text{if } G_i \neq 0 \text{ for all } 1 \leq i \leq m-1. \\ \text{If } G_i = 0 \text{for any } i \text{ from } (1 \leq i \leq m-1), \text{ then} \\ G_{i-1} \text{ - undefined, but } G_{i-2} = q_{i-1}. \end{cases} \tag{2.7}$$

On the basis of sequences $\{\Lambda_i\}(2.6)'$ and $\{G_i\}(2.7)$ we have obtained the following ways**) of computing elements B_{ij} of the matrices $B = C^{-1}$:

Representation 2.2($\Lambda_i \neq 0$ for all $2 \leq i \leq m+1$)

$$B_{ij}(\Lambda) = \begin{cases} B_{ij+1}^{(\Lambda)} \beta_{j+1}, \ 1 \leq j < i \leq m, \\ c_{i+1} B_{i+1j}^{(\Lambda)}, \ 1 \leq i < j \leq m, \end{cases} \tag{2.8}$$

where $B_{ii}(\Lambda) = \Lambda_{i+1}^{-1} + c_{i+1} B_{i+1i+1}^{(\Lambda)} \beta_{i+1}, B_{mm}(\Lambda) = \Lambda_{m+1}^{-1}, \ i = m-1, ..., 1;$

Representation 2.3($G_i \neq 0$ for all $1 \leq i \leq m-1$)

$$B_{ij}(G) = \begin{cases} \hat{c}_i B_{i-1j}^{(G)}, \ 1 \leq j < i \leq m, \\ B_{ij-1}^{(G)} \hat{\beta}_j, \ 1 \leq i < j \leq m, \end{cases} \tag{2.9}$$

where $B_{ii}(G) = G_{i-1}^{-1} + \hat{c}_i B_{i-1i-1}^{(G)} \hat{\beta}_i, B_{11}(G) = G_0^{-1}, \ i = 2, ..., m;$

Representation 2.4(if $\Lambda_j = 0$ for any j from $(2 \leq j \leq m)$ and/or $G_i = 0$ for any i from $(1 \leq i \leq m-1)$, where $j \neq i$)

$$B_{ij}(\Lambda, G) = \begin{cases} \omega_i \prod\limits_{\xi=j+1}^{i} \beta_\xi, \text{ if } 1 \leq j < i \leq m, \\ 0 \text{ for all } i \text{ from } j < i \leq m, \text{ if } \Lambda_j = 0, \\ 0 \text{ for all } j \text{ from } 1 \leq j < i, \text{ if } G_i = 0, \\ \omega_i \prod\limits_{\xi=i+1}^{j} \hat{\beta}_\xi, \text{ if } 1 \leq i < j \leq m, \\ 0 \text{ for all } i \text{ from } 1 \leq i < j, \text{ if } G_j = 0, \\ 0 \text{ for all } j \text{ from } i < j \leq m, \text{ if } \Lambda_i = 0. \end{cases} \tag{2.10}$$

*)When algebraic problems are solved with the use of processes $(2.6)' \div (2.7)$, theoretical and computer zeros are equivalent in nature.

**)In representations given in this paper, use is made of the notation $B_{ij}(\Lambda)$, $B_{ij}(G)$ and $B_{ij}(\Lambda, G)$ for elements B_{ij} to emphasize which sequences $\{\Lambda_i\}$, $\{G_i\}$ or $\{\Lambda_i, G_i\}$ are used to obtain them. Representations 2.2 and 2.3 are employed for inverting matrices C(2.1) with the diagonal dominance.

Diagonal elements B_{ii} of matrices B and the quantities ω_i in (2.10) are defined [1] as follows:

$$\begin{cases} B_{ii} = (\Lambda_{i+1} + G_{i-1} - q_i)^{-1}, \text{ if } \Lambda_i \neq 0 \neq G_i. \\ \text{Here } \omega_i = B_{ii}; \\ B_{ii} = 0, B_{i-1i-1} = G_{i-1}\omega_i, B_{i+1i+1} = G_i^{-1}, \text{ if } \Lambda_i = 0. \\ \text{Here } \omega_i = (-p_i r_i)^{-1}; \\ B_{ii} = 0, B_{i-1i-1} = \Lambda_i^{-1}, B_{i+1i+1} = \Lambda_{i+1}\omega_i, \text{ if } G_i = 0. \\ \text{Here } \omega_i = (-r_{i+1} p_{i+1})^{-1}, \\ i = 1, 2, ..., m. \end{cases} \tag{2.11}$$

In $(2.8) \div (2.11)$ elements Λ and G are defined by $(2.6)' \div (2.7)$, structure elements β, $\hat{\beta}$, $c, \hat{c}$ and their products $\prod \beta_\xi$, $\prod \hat{\beta}_\xi$, $\prod c_\xi$, $\prod \hat{c}_\xi$ are determined [1] simultaneously with Λ and G in the following way:

$$\beta_i = \begin{cases} -p_i \Lambda_i^{-1}, \text{ if } \Lambda_i \neq 0, \\ -p_i, \text{ if } \Lambda_i = 0, \\ \text{and } \beta_{i+1} = -p_{i+1}\omega_i; \end{cases} \quad c_i = \begin{cases} -\Lambda_i^{-1} r_i, \text{ if } \Lambda_i \neq 0, \\ -r_i, \text{ if } \Lambda_i = 0, \\ \text{and } c_{i+1} = -\omega_i r_{i+1}; \end{cases} \tag{2.12}$$

$$\hat{c}_{i+1} = \begin{cases} -G_i^{-1} p_{i+1}, \text{ if } G_i \neq 0, \\ -p_{i+1}, \text{ if } G_i = 0, \\ \text{and } \hat{c}_i = -\omega_i p_i; \end{cases} \quad \hat{\beta}_{i+1} = \begin{cases} -r_{i+1} G_i^{-1}, \text{ if } G_i \neq 0, \\ -r_{i+1}, \text{ if } G_i = 0, \\ \text{and } \hat{\beta}_i = -r_i \omega_i; \end{cases} \tag{2.13}$$

$$\prod_{\xi=j+1}^{i} \beta_\xi = \begin{cases} \beta_i \cdots \beta_{j+1}, \text{ if } j < i, \\ 1, \quad \text{if } j \geq i; \end{cases} \quad \prod_{\xi=i+1}^{j} c_\xi = \begin{cases} c_{i+1} \cdots c_j, \text{ if } i < j, \\ 1, \quad \text{if } i \geq j; \end{cases} \tag{2.14}$$

$$\prod_{\xi=i+1}^{j} \hat{\beta}_\xi = \begin{cases} \hat{\beta}_{i+1} \cdots \hat{\beta}_j, \text{ if } i < j, \\ 1, \quad \text{if } i \geq j; \end{cases} \quad \prod_{\xi=j+1}^{i} \hat{c}_\xi = \begin{cases} \hat{c}_i \cdots \hat{c}_{j+1}, \text{ if } j < i, \\ 1, \quad \text{if} j \geq i. \end{cases} \tag{2.15}$$

Remark 1. The representations $1.2 \div 1.4$ hold valid for the matrix $B = C^{-1}$. Similar representations will take place, evidently, for any $\tilde{B}_\rho^\nu = [C_\rho^\nu]^{-1}$, a submatrix of C_ρ^ν. This fact will be used below in developing the algorithms of solution of the problems $B = C^{-1}$ and $CX = Y$, where*)

$$X = \{x_1, x_2, ..., x_m\}^T, \ Y = \{y_1, y_2, ..., y_m\}^T.$$

Since $B = \overset{\circ}{B} + B_1 Z$ and for Z the matrix equation $\Omega Z = B_2$ holds valid, for ensuring the maximum efficiency (accuracy, computation speed, memory) of computation of B, for $\tilde{B}_\rho^\nu = [C_\rho^\nu]^{-1}$ we employ the representations $2.2 \div 2.4$. Before solving the system $\Omega Z = B_2$, we transform it to the form:

$$\hat{\Omega}\hat{Z} = \hat{B}_2, \text{ where} \\ \hat{\Omega} = \mu^{-1}\Omega\eta^{-1}, \hat{B}_2 = \mu^{-1}B_2, \hat{Z} = \eta Z, \tag{2.16}$$

*)Hereafter T means transposition.

$$\mu = \begin{bmatrix} \begin{bmatrix} \mu_{l_1-1} & \\ & \mu_{l_1} \end{bmatrix} & & & \\ & \begin{bmatrix} \mu_{l_2-1} & \\ & \mu_{l_2} \end{bmatrix} & & \\ & & \cdots\cdots & \\ & & & \begin{bmatrix} \mu_{l_n-1} & \\ & \mu_{l_n} \end{bmatrix} \end{bmatrix}, \quad \eta = \begin{bmatrix} \begin{bmatrix} \eta_{l_1-1} & \\ & \eta_{l_1} \end{bmatrix} & & & \\ & \begin{bmatrix} \eta_{l_2-1} & \\ & \eta_{l_2} \end{bmatrix} & & \\ & & \cdots\cdots & \\ & & & \begin{bmatrix} \eta_{l_n-1} & \\ & \eta_{l_n} \end{bmatrix} \end{bmatrix}. \qquad (2.17)$$

The choice of $\{\mu_{l_i}, \ \eta_{l_i}\}$ – elements of balancing diagonal matrices $\mu, \ \eta$ (2.17) is accomplished in the form

$$\begin{cases} \mu_{l_i-1} = \max\limits_{l_{i-1}+1 \le j \le l_i-2} \left| \prod\limits_{\xi=j+1}^{l_i-1} \beta_\xi \right|, & \eta_{l_i-1} = \max\limits_{l_{i-1}+1 \le j \le l_i-2} \left| \prod\limits_{\xi=j+1}^{l_i-1} c_\xi \right|, \\ \mu_{l_i} = \max\limits_{l_i+1 \le j \le l_{i+1}-2} \left| \prod\limits_{\xi=l_i+1}^{j} \hat{\beta}_\xi \right|, & \eta_{l_i} = \max\limits_{l_i+1 \le j \le l_{i+1}-2} \left| \prod\limits_{\xi=l_i+1}^{j} \hat{c}_\xi \right|, \end{cases} \qquad (2.18)$$

where $i = 1, 2, ..., n, \ \prod \beta_\xi, \ \prod \hat{\beta}_\xi, \ \prod c_\xi, \ \prod \hat{c}_\xi$ are already computed in the process of deriving the system (2.16) products of structure elements $\beta, \hat{\beta}, c, \hat{c}$ (2.12) $\div$ (2.13).

Remark 2. As shown in [3], if the tridiagonal matrix $\hat{\Omega}$ in (2.16) is considered as block–tridiagonal one with dimensions of blocks 2×2, its leading upper (lower) block–angular minors will be nonzero.

Therefore for finding $\hat{Z}$, a solution of the system (2.16)$\div$(2.18) for $(\hat{\Omega}^{-1})_{ij}$, elements of the matrix $\hat{\Omega}^{-1}$ use is made [2] of the following representation:

$$[(\hat{\Omega}^{-1})_{l_i l_j} \equiv \hat{B}_{l_i l_j}] = \begin{cases} \prod\limits_{\xi=l_j+1}^{l_i} [\hat{c}]_\xi [\hat{B}]_{l_j l_j}, \ 1 \le j < i \le n, \\ \prod\limits_{\xi=l_i+1}^{l_j} [c]_\xi [\hat{B}]_{l_j l_j}, \ 1 \le i < j \le n, \end{cases} \qquad (2.19)$$

where $[\hat{B}]_{l_i l_i}$ are diagonal blocks (of dimension 2×2)

$$[\hat{B}]_{l_i l_i} = \begin{bmatrix} \hat{x}_{l_i} & \tilde{r}_{l_i} \\ \tilde{p}_{l_i} & \hat{y}_{l_i} \end{bmatrix}^{-1}, \qquad (2.20)$$

$$\begin{cases} \hat{x}_{l_i} = \tilde{q}_{l_i-1} - \frac{\tilde{p}_{l_i-1} \hat{x}_{l_{i-1}} \tilde{r}_{l_i-1}}{(\tilde{q}_{l_{i-1}} \hat{x}_{l_{i-1}} - \tilde{p}_{l_{i-1}} \tilde{r}_{l_{i-1}})}, \ \hat{x}_{l_1} = \tilde{q}_{l_1-1}, \ i = 2, 3, ..., n; \\ \hat{y}_{l_i} = \tilde{q}_{l_i} - \frac{\tilde{r}_{l_{i+1}-1} \hat{y}_{l_{i+1}} \tilde{p}_{l_{i+1}-1}}{(\tilde{q}_{l_{i+1}-1} \hat{y}_{l_{i+1}} - \tilde{p}_{l_{i+1}} \tilde{r}_{l_{i+1}})}, \ \hat{y}_{l_n} = \tilde{q}_{l_n}, \ i = n-1, ..., 2, 1; \end{cases} \qquad (2.21)$$

$$\begin{cases} \tilde{q}_{l_i-1} = \mu_{l_i-1}^{-1} (q_{l_i-1} - p_{l_i-1} \tilde{B}_{l_i-2 l_i-2} r_{l_i-1}) \eta_{l_i-1}^{-1}, \ \tilde{r}_{l_i} = \mu_{l_i-1}^{-1} r_{l_i} \eta_{l_i}^{-1}; \\ \tilde{p}_{l_i} = \mu_{l_i}^{-1} p_{l_i} \eta_{l_i-1}^{-1}, \ \tilde{q}_{l_i} = \mu_{l_i}^{-1} (q_{l_i} - r_{l_i+1} \tilde{B}_{l_i+1 l_i+1} p_{l_i+1}) \eta_{l_i}^{-1}; \\ \tilde{p}_{l_i-1} = -\mu_{l_i-1}^{-1} (p_{l_i-1} \tilde{B}_{l_i-2 l_{i-1}+1} p_{l_{i-1}+1}) \eta_{l_i-1}^{-1}; \\ \tilde{r}_{l_i-1} = -\mu_{l_i-1}^{-1} (r_{l_{i-1}+1} \tilde{B}_{l_{i-1}+1 l_i-2} r_{l_i-1}) \eta_{l_i-1}^{-1}. \end{cases} \qquad (2.21)'$$

Remark 3. One should remember that the recurrence relations (2.21) are of scalar form, which is a result of that in matrix processes $\{\Lambda_i\}$ and $\{G_i\}$ of the type $(2.6)_{1)}$ and $(2.7)_{1)}$ (see [2]) off-diagonal blocks of the matrix $\hat{\Omega}$(2.16) are of a special form:

$$\begin{bmatrix} 0 & \tilde{p}_{l_i-1} \\ 0 & 0 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 0 \\ \tilde{r}_{l_i-1} & 0 \end{bmatrix}.$$

In this case $(\tilde{q}_{l_{i-1}} \hat{x}_{l_{i-1}} - \tilde{p}_{l_{i-1}} \tilde{r}_{l_{i-1}}) \ne 0$ and $(\tilde{q}_{l_{i+1}-1} \hat{y}_{l_{i+1}} - \tilde{p}_{l_{i+1}} \tilde{r}_{l_{i+1}}) \ne 0$ according to Remark 2 and, as a result, (see [3,4]) $\det([\hat{B}]_{ii}) \ne 0$.

The matrices $[c]_\xi$ and $[\hat{c}]_\xi$ have the form

$$[c]_\xi = \begin{bmatrix} c_{\xi-1} & 0 \\ c_\xi & 0 \end{bmatrix} \text{ and } [\hat{c}]_\xi = \begin{bmatrix} 0 & \hat{c}_{\xi-1} \\ 0 & \hat{c}_\xi \end{bmatrix}.$$

Here $[c_{\xi-1}, c_\xi]^T$ and $[\hat{c}_{\xi-1}, \hat{c}_\xi]^T$ are solutions to the systems of equations:

$$\begin{cases} \begin{bmatrix} \hat{x}_{l_{i-1}} & \tilde{r}_{l_{i-1}} \\ \tilde{p}_{l_{i-1}} & \tilde{q}_{l_{i-1}} \end{bmatrix} \begin{bmatrix} c_{l_i-1} \\ c_{l_i} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{r}_{l_i-1} \end{bmatrix}, \ i = 2, 3, ..., n; \\ \begin{bmatrix} \tilde{q}_{l_{i+1}-1} & \tilde{r}_{l_{i+1}} \\ \tilde{p}_{l_{i+1}} & \hat{y}_{l_{i+1}} \end{bmatrix} \begin{bmatrix} \hat{c}_{l_{i+1}-1} \\ \hat{c}_{l_{i+1}} \end{bmatrix} = \begin{bmatrix} \tilde{p}_{l_{i+1}-1} \\ 0 \end{bmatrix}, \ i = n-1, ..., 2, 1. \end{cases} \qquad (2.22)$$

As we just noted above, determinants $(\tilde{q}_{l_{i-1}} \hat{x}_{l_{i-1}} - \tilde{p}_{l_{i-1}} \tilde{r}_{l_{i-1}}) \ne 0$ and $(\tilde{q}_{l_{i+1}-1} \hat{y}_{l_{i+1}} - \tilde{p}_{l_{i+1}} \tilde{r}_{l_{i+1}}) \ne 0$ of matrices of these systems differ from zero.

Let us finally report the representations obtained in [1] for the solution X of the system $CX = Y$:

$$\begin{cases} x_i = \mathring{x}_i + \bar{c}_i \gamma_k, \ k = n+1, n, ..., 1, \\ i = l_k, l_k - 1, ..., l_{k-1} + 1, \ l_{n+1} = m, \ l_0 = 0, \end{cases} \qquad (2.23)$$

where

$$\begin{cases} \mathring{x}_i = \tilde{B}_{i l_k+1} \tilde{y}_{l_k+1} + \sum\limits_{j=l_k+2}^{l_{k+1}} (\tilde{B}_{ij} y_j), i = l_{k+1}, ..., l_k + 1, \ k = n, ..., 1, 0; \\ [\gamma_k \equiv x_{l_k+1}] = \mathring{x}_{l_k+1} + \bar{c}_{l_k+1} [\gamma_{k+1} \equiv x_{l_{k+1}+1}], [\gamma_{n+1} \equiv x_{m+1}] = 0, \ k = n, ..., 2, 1; \\ \tilde{y}_{l_k+1} = y_{l_k+1} + \bar{\beta}_{l_{k-1}+1} \tilde{y}_{l_{k-1}+1} + \sum\limits_{j=l_{k-1}+2}^{l_k} (\bar{\beta}_j y_j), \tilde{y}_1 = y_1, \ k = 1, 2, ..., n; \\ \bar{\beta}_i = -p_{l_k+1} \tilde{B}_{l_k i}, \bar{c}_i = -\tilde{B}_{i l_k} r_{l_k+1}, \ i = l_{k-1} + 1, ..., l_k, \ k = 1, 2, ..., n. \end{cases} \qquad (2.24)$$

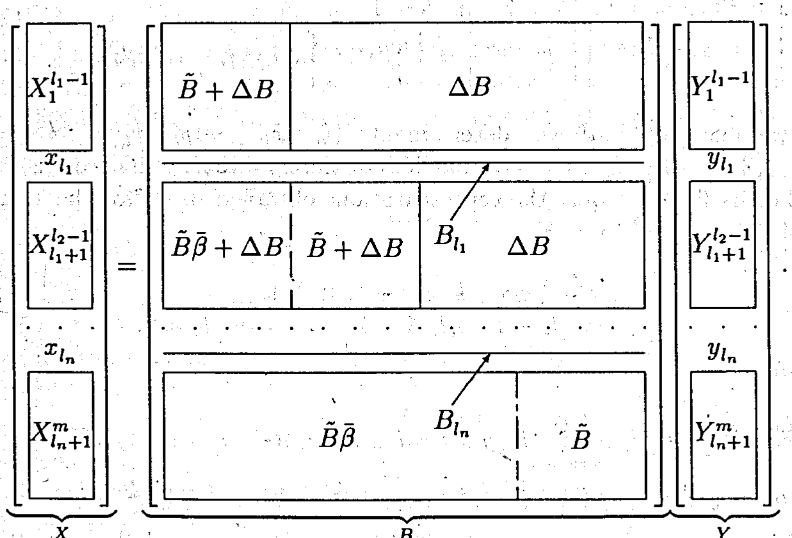
In $(2.23) \div (2.25)$ $\tilde{B}_{ij}$ are elements of submatrices $\tilde{B}^{\nu}_{\rho} = [\tilde{C}^{\nu}_{\rho}]^{-1}$, where

$$\begin{cases} \tilde{C}^{l_{k+1}}_{l_k+1} = C^{l_{k+1}}_{l_k+1} - [0,...,0,p_{l_k+1}]\left[\tilde{C}^{l_k}_{l_{k-1}+1}\right]^{-1}[0,...,0,r_{l_k+1}]^T, \\ \tilde{C}^{l_1}_1 = C^{l_1}_1, k=1,2,...,n, \\ C^{\nu}_{\rho} = \begin{bmatrix} q_\rho & r_{\rho+1} & & & \\ p_{\rho+1} & q_{\rho+1} & r_{\rho+2} & & \\ & \ddots & \ddots & \ddots & \\ & & p_{\nu-1} & q_{\nu-1} & r_\nu \\ & & & p_\nu & q_\nu \end{bmatrix}, \; l_0 = 0, \; l_{n+1} = m. \end{cases} \tag{2.25}$$

For elements of submatrices $\tilde{B}^{\nu}_{\rho} = [\tilde{C}^{\nu}_{\rho}]^{-1}$ the whole Remark 1 holds valid, as well.

Representation $(2.23) \div (2.25)$ was called in [1] the critical–component method of solution of systems of linear equations $CX = Y$.

The graphic scheme corresponding to this representation is of the form [1]:



3. Algorithms of inversion of matrices C and of solution of the system $CX = Y$

In this sect., we present algorithms of inversion of matrices C and of solution of the system of equations $CX = Y$, based on the representations given in sect.2 for $B = C^{-1}$ and X.

Algorithms $I_{a)}$(of inversion of matrices C):

Input data: $\{q_1, q_i, p_i, r_i\}^m_{i=2}$ are elements and m is the dimension of the matrix C; ε is the relative error of computer arithmetic.

Start of computations.

Assignment: $\rho = 0$.

SEtting of the number of the first row (first column) of the submatrix C^{ν}_{ρ}:

$$\rho = \rho + 1. \tag{3.1}$$

Call to the subprogram (algorithm $I_{6)}$) of separation and inversion of the well-posed submatrix C^{ν}_{ρ}.

If $\rho = 1$ and $\nu = m$, computations are completed since B is found.

If $\nu = m$ and $\rho > 1$, tothe algorithm is continued from moment $(3.1)'$.

If $\nu < m$, the sending of elements of the last column and last row $(\tilde{C}^{\nu}_{\rho})^{-1}$ into working arrays for computing the corresponding elements of the matrices B_1, B_2 and Ω.

Assignment:

$$\rho = \nu + 2$$

and return to (3.1).

Computations:

$$\hat{\Omega} = \mu^{-1}\Omega\eta^{-1}, \hat{B}_2 = \mu^{-1}B_2. \tag{3.1$'$}$$

Solution of the system $\hat{\Omega}\hat{Z} = \hat{B}_2$ according to $(2.19) \div (2.22)$.

Obtaining of $B = C^{-1}$:

$$Z = \eta^{-1}\hat{Z}, B = \overset{\circ}{B} + B_1 Z.$$

End of computations.

Algorithm $I_{6)}$(separation and inversion of well-posed submatrices C^{ν}_{ρ}):

Input data:$\{q_{i-1}, q_i, p_i, r_i\}^m_{i=\rho+1}$– elements, m– the dimension of the matrix C and $\rho(m > \rho \geq 1)$ – the number of the first row (first column) of the submatrix C^{ν}_{ρ}; ε – the relative error of computer arithmetic.

Beginning of computations.

Assignments:

$$s = 0; \mu_\rho = 1; \eta_\rho = 1; i = \rho; \Lambda_{i+1} = q_i;$$

$$i = i + 1. \tag{3.2}$$

If $\Lambda_i \neq 0$, the algorithm is continued from moment (3.3), but if $\Lambda_i = 0$, computations

$$\Lambda_{i+1} = (-p_i r_i), \; \beta_i = -p_i$$

and assignment

$$s = 1.$$

If $i \geq m$, algorithm is continued from moment (3.5), but if $i < m$, then the algorithm is continued:

if $\Lambda_{i+1} \neq 0$, algorithm is continued from moment (3.2)′, but if $\Lambda_{i+1} = 0$, computations end up (since $\det(C) = 0$).

Computations:

$$i = i+1, \beta_i = -p_i\Lambda_i^{-1}, \Lambda_{i+1} = q_i; \tag{3.2'}$$

$$\mu_i = \left|\frac{r_i}{p_{i-1}}\right| \max(1, \mu_{i-2}); \;\; \eta_i = \left|\frac{p_i}{r_{i-1}}\right| \max(1, \eta_{i-2})$$

and continuation of the algorithm from moment (3.4).

Computations:

$$\Lambda_{i+1} = q_i - p_i\Lambda_i^{-1}r_i; \;\; c_i = -\Lambda_i^{-1}p_i; \;\; \beta_i = -r_i\Lambda_i^{-1}. \tag{3.3}$$

Assignment:$s = 1$, if $|c_i| > 1$.

If $i \geq m$, the algorithm is continued from moment (3.5), but if $i < m$,, then computations:

$$\mu_i = |\beta_i| \max(1, \mu_{i-1}); \;\; \eta_i = |c_i| \max(1, \eta_{i-1}).$$

Check of the condition of well-posedness) of submatrices C_ρ^ν:*

$$\varepsilon < \mu_i < \varepsilon^{-1} \text{ and/or } \varepsilon < \eta_i < \varepsilon^{-1}. \tag{3.4}$$

If the condition is fulfilled, we return to (3.2), otherwise, we make assignment:

$$i = i - 1;$$

$$\nu = i. \tag{3.5}$$

If $\Lambda_{\nu+1} \neq 0$, the algorithm is continued from moment (3.8) at $s = 1$ and from (3.5)′ at**) $s = 0$, but if $\Lambda_{\nu+1} = 0$, computations end up (since $\det(C) = 0$).

Computation of elements $\tilde{B}_\rho^\nu = (C_\rho^\nu)^{-1}$ (algorithm of representation 1.2):

$$\tilde{B}_{\nu\nu} = \Lambda_{\nu+1}^{-1}; \tag{3.5'}$$

$$j = i; \tag{3.6}$$

*)In essence, conditions (3.4) mean $\varepsilon < |\prod \beta_\xi| < \varepsilon^{-1}$ and $\varepsilon < |\prod c_\xi| < \varepsilon^{-1}$.

**)Here $s = 0$, if C_ρ^ν are submatrices with diagonal dominance of rows.

$$j = j-1, \;\; \tilde{B}_{ij} = \tilde{B}_{ij+1}\beta_{j+1}, \;\; \tilde{B}_{ji} = c_{j+1}\tilde{B}_{j+1i}. \tag{3.7}$$

If $j > \rho$, the return to (3.7), but if $j \leq \rho$, the computations of diagonal elements $\tilde{B}_\rho^\nu$:

$$\tilde{B}_{i-1i-1} = \Lambda_i^{-1} + c_i\tilde{B}_{ii}\beta_i; \;\; i = i-1.$$

If $i > \rho$, the return to (3.6), but if $i \leq \rho$, the end of computations.

Assignments:

$$G_{i-1} = q_i; \tag{3.8}$$

$$i = i - 1. \tag{3.9}$$

If $G_i \neq 0$, the continuation of algorithm from moment (3.10), but if $G_i = 0$, the computations

$$G_{i-1} = (-r_{i+1}p_{i+1}), \hat{\beta}_{i+1} = -r_{i+1}$$

and assignment

$$s = 0.$$

If $i \leq \rho$, continuation of the algorithm from (3.13), but if $i > \rho$, continuation of the algorithm:

if $G_{i-1} \neq 0$, continuation of the algorithm from (3.9)′, but if $G_{i-1} = 0$, the end of computations (since $\det(C) = 0$).

Computations

$$i = i-1, \hat{\beta}_{i+1} = -r_{i+1}G_i^{-1} \tag{3.9'}$$

and return to (3.8).

Computations:

$$G_{i-1} = q_i - r_{i+1}G_i^{-1}p_{i+1}; \;\; \hat{c}_{i+1} = -G_i^{-1}p_{i+1}; \;\; \hat{\beta}_{i+1} = -r_{i+1}G_i^{-1}. \tag{3.10}$$

Assignment $s = 0$, if $|\hat{c}_{i+1}| > 1$.

If $i > \rho$, return to (3.9), but if $i \leq \rho$, continuation of the algorithm:

if $G_{\rho-1} \neq 0$, continuation of the algorithm from moment (3.13) at*) $s = 0$ and from moment (3.10)′ at $s = 1$, but if $G_{\rho-1} = 0$, computations end up (since $\det(C) = 0$).

Computations of elements $\tilde{B}_\rho^\nu = (C_\rho^\nu)^{-1}$ (algorithm of representation 1.3):

$$\tilde{B}_{\rho\rho} = G_{\rho-1}^{-1}; \tag{3.10'}$$

$$j = i; \tag{3.11}$$

*)In this case, if $s = 0$, then C_ρ^ν is a submatrix without diagonal dominance; if $s = 1$, then C_ρ^ν is a submatrix with diagonal dominance of columns.

$$j = j+1,\ \tilde{B}_{ij} = \tilde{B}_{ij-1}\hat{\beta}_j,\ \tilde{B}_{ji} = \hat{c}_j\tilde{B}_{j-1i}. \qquad (3.12)$$

If $j < \nu$, return to (3.12), but if $j \geq \nu$, computations of diagonal elements $\tilde{B}^{\nu}_{\rho}$:

$$\tilde{B}_{i+1i+1} = G_i^{-1} + \hat{c}_{i+1}\tilde{B}_{ii}\hat{\beta}_{i+1};\ i = i+1.$$

If $i < \nu$, the return to (3.11), but if $i \geq \nu$, the end of computations.

Computation of elements $\tilde{B}^{\nu}_{\rho} = (C^{\nu}_{\rho})^{-1}$ *(algorithm of representation 1.4):*

$$i = \nu; \qquad (3.13)$$

$$\begin{cases} \omega_i = (-p_i r_i)^{-1} \text{ and } \tilde{B}_{ii} = 0, \text{ if } \Lambda_i = 0, \\ \omega_i = \Lambda_{i+1}^{-1} \text{ and } \tilde{B}_{ii} = \omega_i, \text{ if } \Lambda_i \neq 0; \end{cases}$$

$$i = i-1. \qquad (3.14)$$

Algorithm of stable computation of $\tilde{B}_{ii}$.

Check of the condition:

$$|\Lambda_i| < |G_i|.$$

If the condition does not hold, continuation of the algorithm from moment (3.14)′, but if the condition is fulfilled, then computations:

$$\begin{cases} \omega_i = (-p_i r_i)^{-1} \text{ and } \tilde{B}_{ii} = 0, \text{ if } \Lambda_i = 0; \\ \omega_i = (\Lambda_{i+1} - r_{i+1}G_i^{-1}p_{i+1})^{-1} \text{ and } \tilde{B}_{ii} = \omega_i, \text{ if } \Lambda_i \neq 0 \end{cases}$$

and continuation of the algorithm from moment (3.15).

Computations:

$$\begin{cases} \omega_i = (-r_{i+1}p_{i+1})^{-1} \text{ and } \tilde{B}_{ii} = 0, \text{ if } G_i = 0; \\ \omega_i = (G_{i-1} - p_i\Lambda_i^{-1}r_i)^{-1} \text{ and } \tilde{B}_{ii} = \omega_i, \text{ if } G_i \neq 0. \end{cases} \qquad (3.14)'$$

Verification of the condition:

$$i > \rho + 1. \qquad (3.15)$$

If the condition is fulfilled, then return to (3.14), otherwise, continuation of the algorithm*):

$$i = i-1;$$

$$\begin{cases} \omega_i = (-r_{i+1}p_{i+1})^{-1} \text{ and } \tilde{B}_{ii} = 0, \text{ if } G_i = 0, \\ \omega_i = G_{i-1}^{-1} \text{ and } \tilde{B}_{ii} = \omega_i \text{ if } G_i \neq 0; \end{cases}$$

$$i = i+1,\ j = i,\ P = 1; \qquad (3.16)$$

*)Here the notation: $P = \prod \beta_\xi$ and $\hat{P} = \prod \hat{\beta}_\xi$. is used.

$$j = j-1,\ P = \beta_{j+1}P,\ \tilde{B}_{ij} = \omega_i P; \qquad (3.17)$$

$$\tilde{B}_{ij} = 0, \text{ if } G_i = 0 \text{ and(or) } \Lambda_j = 0.$$

If $j > \rho$, the return to (3.17), but if $j \leq \rho$, then continuation of computations from moment (3.18) at $i \geq \nu$ and return to (3.16) at $i < \nu$.

Computations:

$$i = i-1,\ j = i,\ \hat{P} = 1; \qquad (3.18)$$

$$j = j+1,\ \hat{P} = \hat{\beta}_j P,\ \tilde{B}_{ij} = \omega_i\hat{P}; \qquad (3.19)$$

$$\tilde{B}_{ij} = 0, \text{ if } \Lambda_i = 0 \text{ and/or } G_j = 0.$$

If $j < \nu$, then return to (3.19), but if $j \geq \nu$, the end of computations at $i \leq \rho$ and return to (3.18) at $i > \rho$.

The end of computations.

Algorithm II(solution of the system $CX = Y$*):*

Input data: $\{q_1, q_i, p_i, r_i; y_1, y_i\}_{i=2}^m$ are elements of the matrix C and the r.h.s. of the system of equations $CX = Y$, respectively; m is the dimension of the matrix C; ε is the relative error of the computer arithmetic.

Start of computations.

Assignments:

$$k = 0,\ i = 1,\ s = 0;$$

$$\Lambda_{i+1} = q_i; \qquad (3.20)$$

$$i = i+1. \qquad (3.21)$$

If $\Lambda_i \neq 0$, then continuation of the algorithm from moment (3.22), but if $\Lambda_i = 0$, then computations:

$$\Lambda_{i+1} = (-p_i r_i);\ \beta_i = -p_i.$$

If $\Lambda_{i+1} \neq 0$, then continuation of the algorithm from moment (3.21)′, but if $\Lambda_{i+1} = 0$, the end of computations (since $\det(C) = 0$).

Assignment:

$$s = 1. \qquad (3.21)'$$

If $i \geq m$, then continuation of the algorithm from moment (3.25), but if $i < m$, then computations

$$i = i+1,\ \beta_i = -p_i\Lambda_i^{-1}$$

and return to (3.20).

Computations:

$$\Lambda_{i+1} = q_i - p_i\Lambda_i^{-1}r_i;\ c_i = -\Lambda_i^{-1}p_i;\ \beta_i = -r_i\Lambda_i^{-1}. \qquad (3.22)$$

Assignment: $s = 1$, if $|c_i| > 1$.

If $i < m$, then return to (3.21), but if $i \geq m$, then continuation of the algorithm: if $\Lambda_{m+1} \neq 0$, then continuation of the algorithm from moment (3.25) at $s = 1$ and from moment (3.22)′ at*) $s = 0$, but if $\Lambda_{m+1} = 0$, then the end of computations (since $\det(C) = 0$).

Obtaining of x_i – components of the solution X by the method of "left run":

$$i = 1, v_1 = y_1/q_1; \tag{3.22'}$$

$$i = i+1, v_i = (y_i - p_i v_{i-1})/\Lambda_{i+1}. \tag{3.23}$$

If $i < m$, then return to (3.23), but if $i \geq m$, then computations:

$$x_m = v_m;$$

$$i = i-1, x_i = c_{i+1}x_{i+1} + v_i. \tag{3.24}$$

If $i > 1$, then return to (3.24), but if $i \leq 1$, then the end of computations.

Assignments:

$$G_{i-1} = q_i; \tag{3.25}$$

$$i = i-1. \tag{3.26}$$

If $G_i \neq 0$, then continuation of the algorithm from moment (3.27), but if $G_i = 0$, then computations:

$$G_{i-1} = (-r_{i+1}p_{i+1});\ \hat{\beta}_{i+1} = -r_{i+1}.$$

If $G_{i-1} \neq 0$, then continuation of the algorithm from moment (3.26)′, but if $G_{i-1} = 0$, then the end of computations (since $\det(C) = 0$).

Assignment

$$s = 0.$$

If $i \leq 1$, then continuation of the algorithm from moment (3.30), but if $i > 1$, then computations

$$i = i-1,\ \hat{\beta}_{i+1} = -r_{i+1}G_i^{-1}$$

and return to (3.25).

Computations:

$$G_{i-1} = q_i - r_{i+1}G_i^{-1}p_{i+1};\ \hat{c}_{i+1} = -G_i^{-1}p_{i+1};\ \hat{\beta}_{i+1} = -r_{i+1}G_i^{-1}. \tag{3.27}$$

Assignment $s = 0$, if $|\hat{c}_{i+1}| > 1$.

*) Here $s = 0$, if C are matrices with diagonal dominance of rows.

If $i > 1$, then return to (3.26), but if $i \leq 1$, then continuation of the algorithm: if $G_0 \neq 0$, continuation of the algorithm from moment (3.30) at*) $s = 0$ and from moment (3.27)′ at $s = 1$, but if $G_0 = 0$, then the end of computations (since $\det(C) = 0$).

Obtaining of x_i – components of the solution X by the method of "right run":

$$i = m,\ w_m = y_m/q_m; \tag{3.27'}$$

$$i = i-1,\ w_i = (y_i - r_{i+1}w_{i+1})/G_{i-1}. \tag{3.28}$$

If $i > 1$, the return to (3.28), but if $i \leq 1$, then computations:

$$x_1 = w_1;$$

$$i = i+1,\ x_i = \hat{c}_i x_{i-1} + w_i. \tag{3.29}$$

If $i < m$, the return to (3.29), but if $i > m$, the end of computations.

*Computation of x_i – components of the solution X (algorithm of representation**) 1.5):*

$$v_1 = y_1,\ i = 1,\ H_1 = 1; \tag{3.30}$$

$$i = i+1,\ \hat{v}_i = (\beta_i H_i/H_{i-1})v_{i-1},\ \bar{v}_i = H_i^{-1}\hat{v}_i; \tag{3.31}$$

$$v_i = \begin{cases} \hat{v}_i, & \text{if } \Lambda_i = 0, \\ \hat{v}_i + H_i y_i, & \text{if } \Lambda_i \neq 0. \end{cases}$$

If $i < m$, the return to (3.31), but if $i \geq m$, continuation of the algorithm:

$$x_m = H_m^{-1}v_m/\Lambda_{m+1},\ w_m = y_m,\ \hat{H}_m = 1;$$

$$i = i-1,\ \hat{w}_i = (\hat{\beta}_{i+1}\hat{H}_i/\hat{H}_{i+1})w_{i+1}; \tag{3.32}$$

$$w_i = \begin{cases} \hat{w}_i, & \text{if } G_i = 0, \\ \hat{w}_i + \hat{H}_i y_i, & \text{if } G_i \neq 0. \end{cases}$$

*) In this case, if $s = 0$, then C is a matrix without diagonal dominance; if $s = 1$, then C is a matrix with diagonal dominance of columns.

**) Here $\{H_i\}$ and $\{\hat{H}_i\}$ are sequences of normalizing factors ensuring the stability of computation of, resp., sequences $\{v_i\}$ and $\{w_i\}$. Here

$$H_1 = 1, \qquad \hat{H}_{l_{k+1}} = 1,$$

$$H_i = \begin{cases} H_{i-1}, & \text{if } |\beta_i| \leq 1, \\ H_{i-1}/|\beta_i|, & \text{if } |\beta_i| > 1, \end{cases} \qquad \hat{H}_i = \begin{cases} \hat{H}_{i+1}, & \text{if } |\hat{\beta}_{i+1}| \leq 1, \\ \hat{H}_{i+1}/|\hat{\beta}_{i+1}|, & \text{if } |\hat{\beta}_{i+1}| > 1. \end{cases}$$

Note that in the programs described below we took $\{H_i = 1\}_{i=1}^{m}, \{\hat{H}_i = 1\}_{i=l_k+1}^{l_{k+1}}, k = n, ..., 1, 0, l_0 = 0, l_{n+1} = m$.

If $i > 1$, continuation of the algorithm from moment (3.33), but if $i \leq 1$, then

$$x_1 = (\hat{H}_1^{-1} w_1)/\mathrm{G}_0.$$

and the end of computations.

Algorithm of stable computation of B_{ii}.

Check of the condition:

$$|\Lambda_i| < |\mathrm{G}_i|. \tag{3.33}$$

If the condition is not fulfilled, the continuation of the algorithm from moment (3.34), and when the condition holds valid, then computations

$$\begin{cases} \omega_i = (-p_i r_i)^{-1} \text{ and } \bar{w}_i = 0, \text{ if } \Lambda_i = 0, \\ \omega_i = (\Lambda_{i+1} - r_{i+1}\mathrm{G}_i^{-1}p_{i+1})^{-1} \text{ and } \bar{w}_i = w_i\hat{H}_i^{-1}, \text{ if } \Lambda_i \neq 0 \end{cases}$$

and continuation of the algorithms from moment (3.35).

Computations:

$$\omega_i = \begin{cases} (-r_{i+1}p_{i+1})^{-1}, \text{ if } \mathrm{G}_i = 0, \\ (\mathrm{G}_{i-1} - p_i\Lambda_i^{-1}r_i)^{-1}, \text{ if } \mathrm{G}_i \neq 0. \end{cases} \tag{3.34}$$

$$\bar{w}_i = w_i\hat{H}_i^{-1}.$$

Computation of x_i – well-posed components; separation and computation of x_{l_k} – critical components of the solution X

$$x_i = \omega_i(\bar{v}_i + \bar{w}_i). \tag{3.35}$$

Assignments:

$$\begin{cases} F_{i+1} = |y_{i+1}| \text{ and } f_{i+1} = 1, \text{ if } |y_{i+1}| > 1 \text{ or} \\ F_{i+1} = 1 \text{ and } f_{i+1} = |y_{i+1}|, \text{ if } |y_{i+1}| \leq 1. \end{cases}$$

Verification of the condition of "extended discrepancy":

$$\left|F_{i+1} - |p_{i+1}x_i + q_{i+1}x_{i+1} + r_{i+2}x_{i+2}|/f_{i+1}\right| < 2\varepsilon.$$

If the condition is valid, then return to (3.32), but if it is not fulfilled, , then computations:

$$k = k + 1, \; l_k = i; \tag{3.36}$$

$$\bar{w}_i = w_i = y_i, \; \varphi_i = -r_{i+1}, \mathrm{G}_{i-1} = q_i, \omega_i = \Lambda_{i+1}^{-1}, \hat{H}_i = 1;$$

$$\overset{\circ}{x}_i = \bar{v}_i + \bar{w}_i, \; x_i = \omega_i(\overset{\circ}{x}_i + \varphi_i x_{l_k+1}). \tag{3.37}$$

If $i > 1$, continuation of the algorithm from moment (3.38), but if $i \leq 1$, the end of computations.

Assignment:

$$i = i - 1. \tag{3.38}$$

If $\mathrm{G}_i \neq 0$, continuation of the algorithm from moment (3.41), but if $\mathrm{G}_i = 0$, then computations:

$$\mathrm{G}_{i-1} = (-r_{i+1}p_{i+1}); \; \hat{\beta}_{i+1} = -r_{i+1}; \; w_i = (\hat{\beta}_{i+1}\hat{H}_i/\hat{H}_{i+1})w_{i+1}.$$

If $\mathrm{G}_{i-1} \neq 0$, continuation of the algorithm from moment (3.39), but if $\mathrm{G}_{i-1} = 0$, then the end of computations (since $\det(C) = 0$).

Computations:

$$x_i = (\hat{H}_i^{-1}w_i)/\mathrm{G}_{i-1}. \tag{3.39}$$

If $i > 1$, continuation of the algorithm from moment (3.40), but if $i \leq 1$, then the end of computations.

Computations:

$$i = i - 1, \; \hat{\beta}_{i+1} = -r_{i+1}\mathrm{G}_i^{-1}, \varphi_i = \frac{r_{i+1}}{p_i}\varphi_{i+2} \tag{3.40}$$

and continuation of the algorithm from moment (3.42).

Computations:

$$\mathrm{G}_{i-1} = q_i - r_{i+1}\mathrm{G}_i^{-1}p_{i+1}, \; \hat{\beta} = -r_{i+1}\mathrm{G}_i^{-1}, \; \varphi_i = \hat{\beta}_{i+1}\varphi_{i+1}. \tag{3.41}$$

Check of the condition) of well - posedness of submatrices $\tilde{C}_\rho^\nu$:*

$$\varepsilon < |\varphi_i| < \varepsilon^{-1}. \tag{3.42}$$

If the condition does not hold valid, return to (3.36), but if it is valid, then computation

$$w_i = (\hat{\beta}_{i+1}\hat{H}_i/\hat{H}_{i+1})w_{i+1} + \hat{H}_i y_i.$$

Algorithm of stable computation of $\tilde{B}_{ii}$.

Verification of the condition:

$$|\Lambda_i| < |\mathrm{G}_i|.$$

If the condition is not fulfilled, continuation of the algorithm from moment (3.43), otherwise, computations

$$\begin{cases} \omega_i = (-p_i r_i)^{-1} \text{ and } \bar{w}_i = 0, \text{ if } \Lambda_i = 0, \\ \omega_i = (\Lambda_{i+1} - r_{i+1}\mathrm{G}_i^{-1}p_{i+1})^{-1} \text{ and } \bar{w}_i = w_i\hat{H}_i^{-1}, \text{ if } \Lambda_i \neq 0 \end{cases}$$

---

*) The condition (3.42) essentially means that $\varepsilon < |\prod \hat{\beta}_\xi| < \varepsilon^{-1}$.

and return to (3.37).

Computations:

$$\omega_i = (G_{i-1} - p_i\Lambda_i^{-1}r_i)^{-1},\ \tilde{\omega}_i = w_i\hat{H}_i^{-1} \tag{3.43}$$

and return to (3.37).

The end of computations.

Below, we present block-schemes of the programs of the described algorithms.

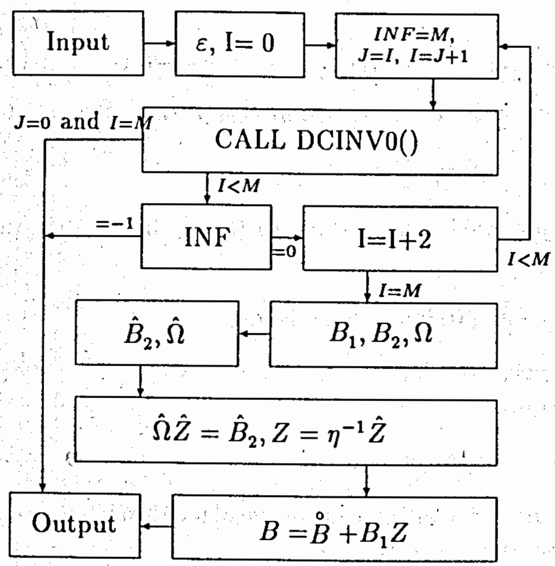


Fig.1. Block-scheme of the program DCINV1 of the algorithm $I_{a)}$.

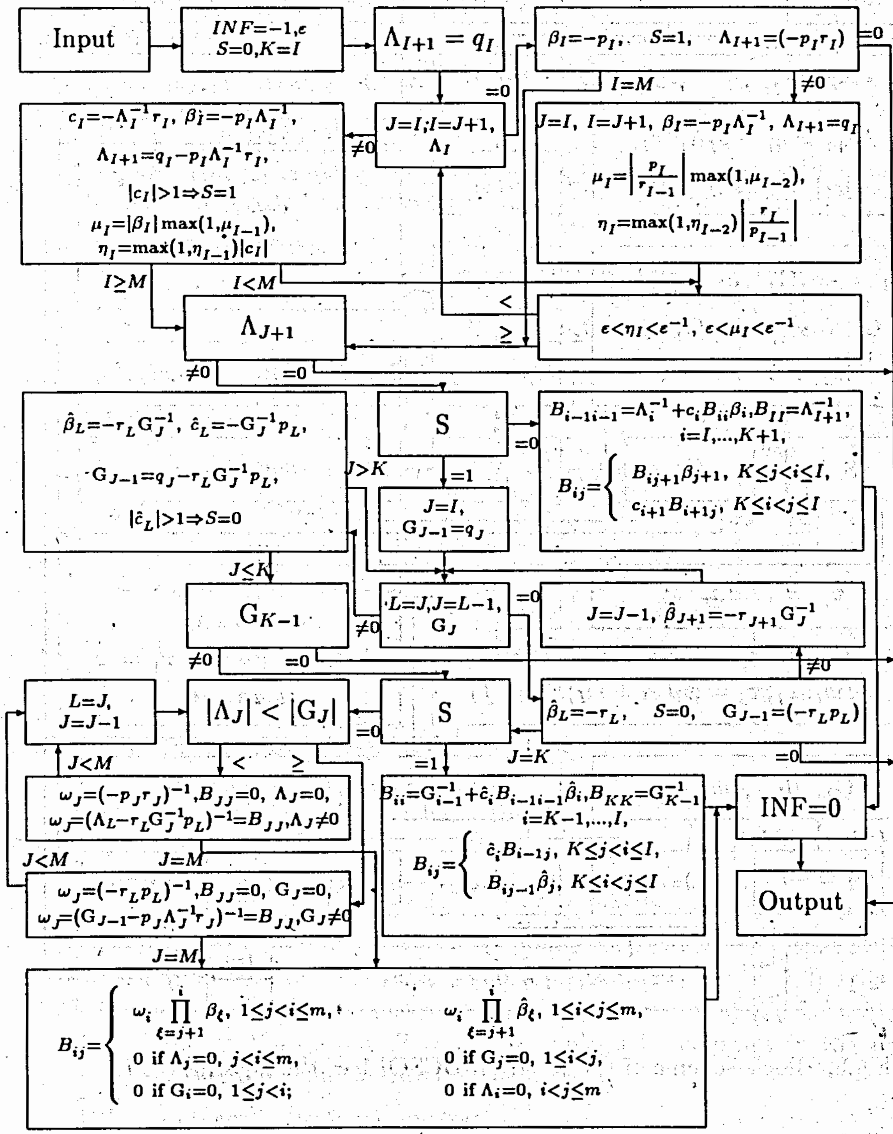


Fig2. Block-scheme of the program DCINV0 of the algorithm $I_{6)}$.

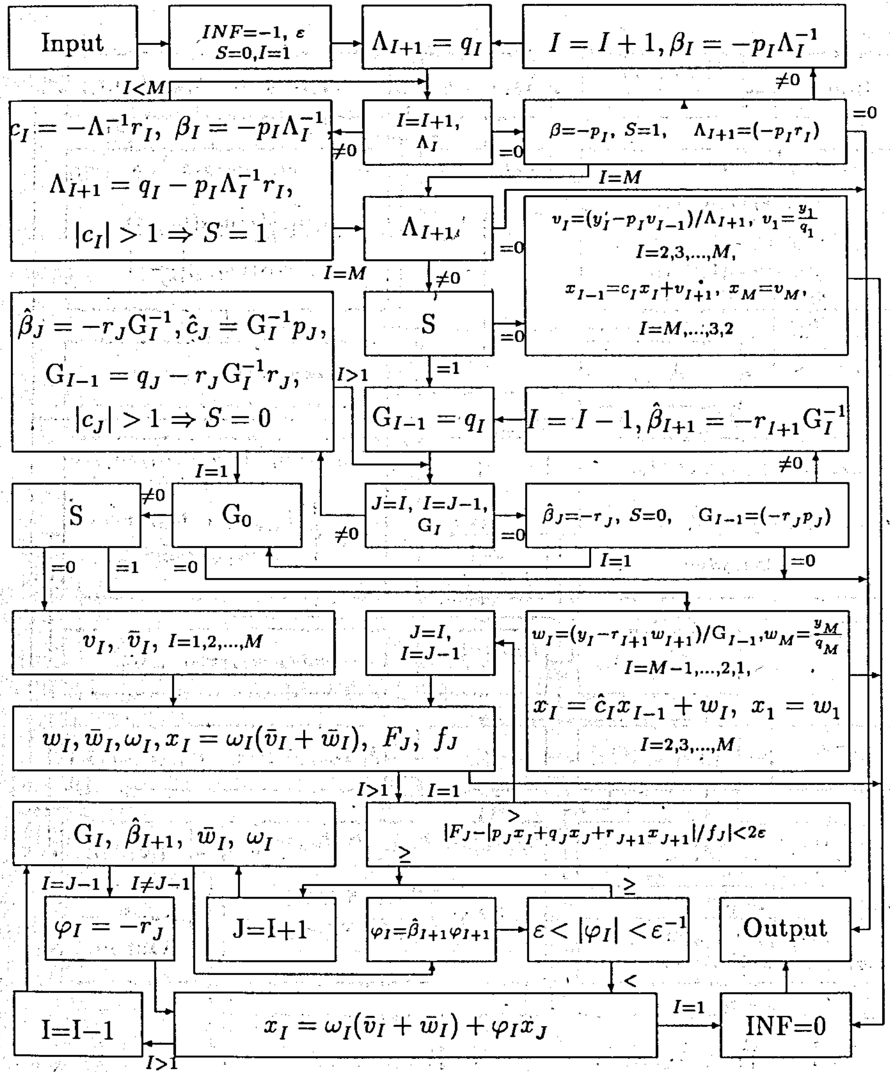


Fig.3. Block–scheme of the program DCSOL1 of the algorithm II.

4. Description of programs of inversion of matrices C and solution of systems of linear equations $CX = Y$

In this sect., we describe the program DCINV1 – inversion of matrices C (2.1) of the general form and the program DCSOL1 – solution of systems $CX = Y$. The programs are written in FORTRAN–77 on the basis of algorithms $I_{a)}$, $I_{b)}$ and II presented in sect.3.

I. Program DCINV1.

Program DCINV1 inverts the matrix $C(2.1)$ by using the subprogram DCINV0 that separated in array A and inverts well–posed submatrices C_ρ^ν:

$$C_\rho^\nu = \begin{bmatrix} * & q_\rho & r_{\rho+1} \\ p_{\rho+1} & q_{\rho+1} & r_{\rho+2} \\ \vdots & \vdots & \vdots \\ p_{\nu-1} & q_{\nu-1} & r_\nu \\ p_\nu & q_\nu & * \end{bmatrix} \Leftarrow \left\{ \left[\overbrace{\begin{bmatrix} * & q_1 & r_2 \\ p_2 & q_2 & r_3 \\ \vdots & \vdots & \vdots \\ p_\rho & q_\rho & r_{\rho+1} \\ \vdots & \vdots & \vdots \\ p_\nu & q_\nu & r_{\nu+1} \\ \vdots & \vdots & \vdots \\ p_{M-1} & q_{M-1} & r_M \\ p_M & q_M & * \end{bmatrix}}^{C} \begin{matrix} *\cdots* \\ *\cdots* \\ \vdots \\ *\cdots* \\ \vdots \\ *\cdots* \\ \vdots \\ *\cdots* \\ *\cdots* \end{matrix} \right] \begin{matrix} \leftarrow 1 \\ \\ \\ \leftarrow \rho \\ = A, \\ \leftarrow \nu \\ \\ \\ \leftarrow M \end{matrix} \right. \qquad (4.1)$$

where symbol $*$ marks cells of array A, whose contents is not important at the input.

Call to the subprogram DCINV1:

```
CALL DCINV1(M,A,INF,R1,R2,R,IZ)
```

Here:

M – (integer) dimension of the quadratic matrix C (2.1);

A – (real*8) a two-dimensional array of dimension $M \times M$, that at the input contains the matrix C (2.1), and at the output*) $B = C^{-1}$. At the input, the matrix C lies in the array A in the form (4.1) and at the output $B = C^{-1}$ is located by columns in the first M–columns and M–rows of the array A;

INF – (integer) output parameter:

*)The described version of program DCINV1 assumes the storage of $B = C^{-1}$ in the IBM memory. However, the program can easily be modified also for storage of B elements on an external medium.

INF= 0 – normal completion of the program work;

INF= –1 – initial matrix is singular;

R1,R2,R – (real*8) one-dimensional working arrays of dimension M;

IZ – (integer) one-dimensional working array of dimension M.

II. Subprogram DCINV0.

The subprogram DCINV0 works in two regimes: in the regime together with DCINV1, that itself calls to DCINV0, and in an independent regime*) In the latter case, call to DCINV0 is of the form:

CALL DCINV0(M,A,I,INF,R1,R2,R)

Here: M, A and R1 are the same parameters as in the program DCINV1.

I,INF**) – (integer)
at the input: I=1, INF=M. Here I and INF are assigned before call to the subprogram DCINV0;
at the output: I=M and

INF= 0 – normal completion of the work of subprograms;

INF= –1 – the initial matrix $C(2.1)$ is singlet;

R,R2 – (real*8) one-dimensional working arrays of dimension M. At the output R and R2 contain***) elements of sequences $\{\Lambda_i\}$ and $\{G_i\}$.

III. Program DCSOL1.

Program DCSOL1 solves the system of linear equations $CX = Y$ with the tridiagonal matrix $C(2.1)$.

Call to the subprogram DCSOL1:

CALL DCSOL1(M,A,INF,IDIM,B,R1,R2,R)

Here:

*)If it is a priori known that the initial matrix $C(2.1)$ is well-posed, then it can be inverted immediately by DCINV0, without call to DCINV1.

**)If DCINV0 works in the regime with DCINV1, then at the input in DCINV0: I= ρ, INF= ν; at the output, I= ν.

***)At the output in the regime with DCINV1:

$$R:\{\ldots,\eta_{\rho-1},\Lambda_{\rho+1},\Lambda_{\rho+2},\ldots,\Lambda_{i-1},0,\omega_i^{-1},\Lambda_{i+2},\ldots,\Lambda_{\nu+1},\eta_{\nu+1},\ldots\};$$

$$R2:\{\ldots,\mu_{\rho-1},G_{\rho-1},\ldots,G_{i-2},\omega_i^{-1},0,G_{i+1},\ldots,G_{\nu-2},G_{\nu-1},\mu_{\nu+1},\ldots\}.$$

M – (integer) dimension of the matrix $C(2.1)$;

A – (real*8) a two-dimensional array of dimension M×3 that at the input contains the matrix $C(2.1)$ in the form:

$$A=\begin{bmatrix} * & q_1 & r_2\\ p_2 & q_2 & r_3\\ \vdots & \vdots & \vdots\\ p_{M-1} & q_{M-1} & r_M\\ p_M & q_M & * \end{bmatrix};\quad B=\begin{bmatrix} b_{11} & b_{12} & \cdots b_{1IDIM}\\ b_{21} & b_{22} & \cdots b_{2IDIM}\\ \vdots & \vdots & \vdots\\ b_{M1} & b_{M2} & \cdots b_{MIDIM}\end{bmatrix};\quad (4.2)$$

INF – (integer) an output parameter:

INF= 0 – normal completion of the work of subprograms;

INF= –1 – the initial matrix is singular;

IDIM – (integer) the number of r.h.s. of the system $CX = Y$;

B – (real*8) a two-dimensional array of dimension M×IDIM (B can be one-dimensional if IDIM=1) containing:
at the input, the matrix of r.h.s. in the form (4.2);
at the output, the matrix of solutions in the form (4.2)

R1,R2,R – (real*8) one-dimensional working arrays of dimension M. At the output arrays R and R2 contain elements of sequences $\{\Lambda_i\}$ and $\{G_i\}$ in the form:

$$R:\{\Lambda_2,\Lambda_3,\ldots,\Lambda_{i-1},0,\omega_i^{-1},\Lambda_{i+2},\ldots,\Lambda_{M+1}\};$$

$$R2:\{G_0,\ldots,G_{i-2},\omega_i^{-1},0,G_{i+1},\ldots,G_{M-2},G_{M-1}\}.$$

Remark 4. Subprograms DCINV1, DCINV0 and DCSOL1 use the block COMMON Q1, DET. In Q1 we define $\varepsilon = \beta^{-t}$ – the relative error of computer arithmetic*). When the work of programs DCINV1 and DCSOL1 is completed, in Q1 we find the value of ε and in DET the value of the determinant of the matrix C.

*)The value of ε: $\varepsilon = 16^{-13}$ – at IBM of series EC; $\varepsilon = 2^{-55}$ – at IBM of the type CM-4; $\varepsilon = 2^{-52}$ – at IBM PC. If ε is unknown, it can be obtained by using our subprogram-function UNDFLQ:

EPS=UNDFLQ(BETA),

where BETA – (integer) the number given at the input whose value is ($\beta \geq 2$) the basis of the number system of a given IBM. The value of ε can also be determined with the use of the subprogram INIT [5].

5. Results of numerical experiments and their analysis

In this sect., we discuss the results of numerical experiments on testing the above programs of inversion of matrices C and of solution of systems of linear equations $CX = Y$ using various particular examples taken, in particular, from [6,7].

Let us first explain the notation and abbreviations adopted in Tables $1 \div 6$:

M – the dimension of matrix C;

Cond(C) – the condition number of matrix C;

$\|E-BC\|_E$ – the Euclidean norm of the right discrepancy in inverting matrices C;

$\|E-CB\|_E$ – the Euclidean norm of the left discrepancy in inverting matrices C;

$D1=\|Y-C\tilde{X}\|_E$ – the Euclidean norm of discrepancy;

$D2=\left|\|X\|_E-\|\tilde{X}\|_E\right|$ – modulus of the difference of Euclidean norms of X – exact and $\tilde{X}$ – approximate solutions of the system $CX = Y$;

D1+D2 – the sum of the discrepancy norm and modulus of the norm difference between X and $\tilde{X}$;

$\|\tilde{X}\|_E$ – the Euclidean norm of $\tilde{X}$.

Notation of subprograms in Tables (OU – our subprogram):

OU – subprogram of DCINV1 in Tables 1,2 and of DCSOL1 in Tables $3 \div 6$;

GS – subprogram of DBEQN from library CERNLIB;

TL – subprogram of DTSYS from library LIBJINR;

QR – subprograms of (F01AXF and F04AHF) from library NAGLIB;

ML – subprogram–function of PSOL from LINAa [5];

TH – subprogram of SLAY from library LIBJINR.

Here: CERNLIB – libraries of subprograms of CERN [8]; NAGLIB – package of mathematical programs (Numerical Algorithms Group, Oxford) [9]; LIBJINR – JINR library of programs [10]; LINA – package of programs [5].

Our subprograms of DCINV0, DCINV1 and DCSOL have realized the algorithms suggested in the present paper.

The subprogram of DBEQN realized the modified algorithm of the Gauss exclusion method.

The subprogram of DTSYS employs the algorithm [11] of the nonmonotone orthogonal - run method.

Subprograms of F01AXF and F04AHF are based on algorithms of the QR–method.

Subprogram of PSOL realized the algorithm of the singular–expansion method with the use of exhaustion.

Subprogram of SLAY realized algorithms of the Tikhonov regularization method.

Note also that the Euclidean norms $\|\cdot\|_E$ were computed with the use of our subprograms of TOCHI3 and TOCHS3. Com.time – the computation time is obtained with the help of subprograms TIMEST and TIMEX from the library CERNLIB. The Cond(C) was computed by using the subprogram of PSOL from the package of LINA. When Cond(C)$> 1/\varepsilon$, where ε is the relative accuracy of computer arithmetic, in Tables Cond(C)= 0 [5] is presented.

Tables do not contain values of Com.time higher than 100 sec.

Subprogram of SLAY worked up to M=100.

So, in Tables 1,2 the results of inverting matrices C are reported; in Tables $3 \div 6$, the results of solution of the system $CX = Y$.

All the programs given above have been tested with specific testing examples listed below. Examples 1,2,3,5 are taken from [6,7]. They are usually used in testing programs of the inversion of matrices C and solution of systems $CX = Y$. As an extra example, we propose testing example 4.

In Tables $1 \div 6$, for comparison, we present the results of computations with all listed programs for the most ill–posed matrices and systems of equations.

Solutions of the systems of equations $CX = Y$ derived by our methods obey [3] the conditions: $\min_X \|CX - Y\|$ и $\min_X \|X\|$.

I. Testing examples of matrices C (for inversion):

Example 1

$$C = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{bmatrix};$$

Example 2

$$C = \begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & a \end{bmatrix},$$

where $a = \frac{1-M}{M}$;

Example 3

$$C = \begin{bmatrix} 1 & -1 & & & \\ -1 & 1 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 1 & -1 \\ & & & -1 & 1 \end{bmatrix};$$

Example 4

$$C = \begin{bmatrix} 1 & r & & & \\ p & 1 & r & & \\ & \ddots & \ddots & \ddots & \\ & & p & 1 & r \\ & & & p & 1 \end{bmatrix},$$

where $p = 1 + \varepsilon_0,\ r = 1 - \varepsilon_0,\ \varepsilon_0 = 0,0000001$;

Example 5

$$C = \begin{bmatrix} 6 & 3 & & & \\ 4 & 6 & 3 & & \\ & \ddots & \ddots & \ddots & \\ & & 4 & 6 & 3 \\ & & & 4 & 6 \end{bmatrix}.$$

Table 1(Results of computations of example 2)

	No pr.	Com.time (sec.)	$\|\|E-BC\|\|$	$\|\|E-CB\|\|$
	OU	0.0015	0.109E-13	0.109E-13
M=10	GS	0.0020	0.109E-13	0.109E-13
	TL	0.0048	0.109E-13	0.109E-13
	QR	0.0336	0.263E-13	0.513E-13
Cond(A)=0.289E03	ML	1.3506	0.633E-13	0.560E-13
	TH	0.2452	0.883E-12	0.313E-13
	OU	0.0218	0.000E+00	0.000E+00
M=50	GS	0.0394	0.000E+00	0.000E+00
	TL	0.1213	0.000E+00	0.000E+00
	QR	6.2447	0.267E-11	0.189E-11
Cond(A)=0.704E04	ML	5.5441	0.381E-11	0.339E-11
	TH	89.7781	0.359E-09	0.333E-11
	OU	0.0759	0.000E+00	0.000E+00
M=100	GS	0.1555	0.000E+00	0.000E+00
	TL	0.4871	0.000E+00	0.000E+00
	QR	85.7524	0.172E-10	0.736E-11
Cond(A)=0.280E05	ML	30.4805	0.212E-10	0.268E-10
	TH		0.421E-08	0.217E-10
	OU	0.2822	0.651E-11	0.651E-11
M=200	GS	0.6394	0.651E-11	0.651E-11
	TL	2.0646	0.651E-11	0.651E-11
	QR		0.133E-09	0.818E-10
Cond(A)=0.112E06	ML		0.147E-09	0.171E-09
	TH			
	OU	0.6328	0.000E+00	0.000E+00
M=300	GS	1.5934	0.000E+00	0.000E+00
	TL	4.4904	0.000E+00	0.000E+00
	QR		0.347E-09	0.217E-09
Cond(A)=0.251E06	ML		0.473E-09	0.448E-09
	TH			
	OU	1.2187	0.000E+00	0.000E+00
M=400	GS	2.6508	0.000E+00	0.000E+00
	TL	8.0338	0.000E+00	0.000E+00
	QR			
Cond(A)=0.000E00	ML			
	TH			
	OU	1.9312	0.000E+00	0.000E+00
M=500	GS	4.5707	0.000E+00	0.000E+00
	TL	12.5360	0.000E+00	0.000E+00
	QR			
Cond(A)=0.000E00	ML			
	TH			

Table 2(Results of computations of example 5)

	No pr.	Com.time (sec.)	$\|\|E-BC\|\|$	$\|\|E-CB\|\|$
	OU	0.0027	0.375E-14	0.384E-14
M=10	GS	0.0021	0.706E-14	0.233E-14
	TL	0.0051	0.104E+02	0.332E+01
	QR	0.0336	0.184E-13	0.145E-13
Cond(A)=0.107E03	ML	1.3534	0.273E-13	0.192E-13
	TH	0.2448	0.105E-12	0.161E-13
	OU	0.0442	0.146E-11	0.541E-12
M=50	GS	0.0358	0.453E-11	0.457E-12
	TL	0.1268	0.263E-11	0.701E-12
	QR	6.2988	0.336E-11	0.546E-11
Cond(A)=0.186E05	ML	5.2973	0.560E-11	0.600E-11
	TH	90.7625	0.790E-08	0.420E-11
	OU	0.1675	0.416E-08	0.170E-08
M=100	GS	0.1548	0.691E-08	0.111E-08
	TL	0.5164	0.592E-08	0.123E-08
	QR	86.6941	0.134E-07	0.169E-07
Cond(A)=0.494E08	ML	30.8325	0.251E-07	0.183E-07
	TH		0.556E-01	0.103E-07
	OU	0.6509	0.204E-02	0.166E-02
M=200	GS	0.6132	0.164E-01	0.164E-02
	TL	2.0562	0.861E-02	0.135E-02
	QR		0.190E-01	0.435E-01
Cond(A)=0.437E14	ML		0.248E-01	0.139E-01
	TH			
	OU	1.5156	0.272E+05	0.475E+04
M=300	GS	1.4090	0.517E+05	0.759E+04
	TL	4.4898	0.151E+05	0.718E+04
	QR		0.366E+03	0.250E+03
Cond(A)=0.000E00	ML		0.101E+06	0.565E+05
	TH			
	OU	2.8059	0.782E+11	0.728E+10
M=400	GS	2.7754	0.158E+12	0.128E+11
	TL	8.2067	0.713E+11	0.106E+11
	QR			
Cond(A)=0.000E00	ML			
	TH			
	OU	4.3426	0.685E+17	0.109E+17
M=500	GS	4.1601	0.157E+18	0.979E+16
	TL	12.6268	0.437E+17	0.901E+16
	QR			
Cond(A)=0.000E00	ML			
	TH			

From the general analysis of the results of testing (in particular, given in Tables 1 and 2) it follows that our program OU (DCINV1) has on the average better time and accuracy characteristics.

II. Testing examples of systems of linear equations $CX = Y$ with tridiagonal matrices of the general form:

System 1

C-matrix of example 1,

$x_i = \frac{1}{i},\ i = 1,2,...,M,\ y_1 = 2,\ y_M = \frac{M-2}{M(M-1)},$

$y_i = \frac{2}{(1-i)i(1+i)},\ i = 2,3,...,M-1;$

System 2

C-matrix of example 2,

$x_i = 1 + (-1)^i\varepsilon_0,\ i = 1,2,...,M,$

$y_1 = 2\varepsilon_0,\ y_i = (-1)^{i-1}4\varepsilon_0,\ i = 2,3,...,M-1,$

$y_M = (-1)^{M-1}(a+\varepsilon_0)\varepsilon_0,\ a = \frac{1-M}{M},\ \varepsilon_0 = 0,00000001;$

System 3

C-matrix of example 3,

$x_i = \frac{1}{2i},\ i = 1,2,...,M,\ y_1 = \frac{1}{4},\ y_M = \frac{1}{2M(1-M)},$

$y_i = \frac{i^2+1}{2i(1-i)(1+i)},\ i = 2,3,...,M-1;$

System 4

C-matrix of example 4,

$x_i = 1,\ i = 1,2,...,M,\ y_1 = \varepsilon_0,\ y_M = 2+\varepsilon_0,$

$y_i = 3,\ i = 2,3,...,M-1,\ \varepsilon_0 = 0,0000001;$

System 5

C-matrix of example 5,

$x_i = 1,\ i = 1,2,...,M,$

$y_1 = 9, y_M = 10,\ y_i = 13,\ i = 2,3,...,M-1.$

Table 3(Results of computations of system 3)

	No pr.	Com.time (sec.)	D1= $\|\|Y - C\tilde{X}\|\|$	D2= $\|\|\|X\|\| - \|\|\tilde{X}\|\|\|$	D2+D1	$\|\|\tilde{X}\|\|$
	OU	0.0013	0.000E+00	0.000E+00	0.000E+00	0.622E+00
M=10	GS	0.0007	0.159E-16	0.139E-15	0.155E-15	0.622E+00
	TL	0.0008	0.000E+00	0.000E+00	0.000E+00	0.622E+00
Cond(C)	QR	0.0037	0.278E-15	0.139E-15	0.417E-15	0.622E+00
=.173E02	ML	0.1307	0.454E-15	0.111E-15	0.565E-15	0.622E+00
	TH	0.0247	0.304E+01	0.221E+01	0.524E+01	0.283E+01
	OU	0.0051	0.139E-16	0.139E-15	0.153E-15	0.637E+00
M=51	GS	0.0025	0.239E-16	0.139E-15	0.163E-15	0.637E+00
	TL	0.0040	0.139E-16	0.139E-15	0.153E-15	0.637E+00
Cond(C)	QR	0.1342	0.601E-15	0.139E-15	0.740E-15	0.637E+00
=.864E02	ML	3.8904	0.155E-14	0.625E-15	0.218E-14	0.637E+00
	TH	1.9282	0.721E+01	0.636E+01	0.136E+02	0.700E+01
	OU	0.0098	0.000E+00	0.111E-15	0.111E-15	0.639E+00
M=100	GS	0.0045	0.239E-16	0.250E-15	0.274E-15	0.639E+00
	TL	0.0114	0.000E+00	0.111E-15	0.111E-15	0.639E+00
Cond(C)	QR	0.8579	0.922E-15	0.347E-15	0.127E-14	0.639E+00
=.167E03	ML	18.6607	0.239E-14	0.178E-14	0.417E-14	0.639E+00
	TH	13.6433	0.101E+02	0.926E+01	0.193E+02	0.990E+01
	OU	0.0194	0.139E-16	0.139E-15	0.153E-15	0.640E+00
M=201	GS	0.0085	0.240E-16	0.139E-15	0.163E-15	0.640E+00
	TL	0.0158	0.139E-16	0.139E-15	0.153E-15	0.640E+00
Cond(C)	QR	6.8556	0.165E-14	0.666E-15	0.232E-14	0.640E+00
=.335E03	ML		0.359E-14	0.210E-14	0.569E-14	0.640E+00
	TH					
	OU	0.0283	0.139E-16	0.139E-15	0.153E-15	0.641E+00
M=300	GS	0.0123	0.241E-16	0.139E-15	0.163E-15	0.641E+00
	TL	0.0226	0.139E-16	0.139E-15	0.153E-15	0.641E+00
Cond(C)	QR	22.3475	0.245E-14	0.501E-14	0.746E-14	0.641E+00
=.498E03	ML		0.393E-14	0.472E-14	0.865E-14	0.641E+00
	TH					0.641E+00
	OU	0.0537	0.000E+00	0.000E+00	0.000E+00	0.500E-01
M=400	GS	0.0163	0.875E-17	0.195E-15	0.204E-15	0.500E-01
	TL	0.0237	0.000E+00	0.000E+00	0.000E+00	0.500E-01
Cond(C)	QR	50.8679	0.291E-15	0.457E-15	0.748E-15	0.500E-01
=.663E03	ML		0.695E-15	0.728E-15	0.142E-14	0.500E-01
	TH					
	OU	0.0486	0.000E+00	0.000E+00	0.000E+00	0.447E-01
M=501	GS	0.0197	0.419E-17	0.220E-15	0.225E-15	0.447E-01
	TL	0.0309	0.000E+00	0.000E+00	0.000E+00	0.447E-01
Cond(C)	QR	99.5615	0.289E-15	0.466E-15	0.755E-15	0.447E-01
=.831E03	ML		0.591E-15	0.533E-15	0.112E-14	0.447E-01
	TH					

Table 4(Results of computations of system 4)

	No pr.	Com.time (sec.)	D1= $\|\|Y - C\tilde{X}\|\|$	D2= $\|\|\|X\|\| - \|\|\tilde{X}\|\|\|$	D2+D1	$\|\|\tilde{X}\|\|$
	OU	0.0014	0.628E-15	0.000E+00	0.628E-15	0.316E+01
M=10	GS	0.0008	0.942E-15	0.222E-15	0.116E-14	0.316E+01
	TL	0.0009	0.186E-01	0.363E-03	0.190E-01	0.316E+01
Cond(C)	QR	0.0037	0.140E-14	0.222E-15	0.163E-14	0.316E+01
=.173E02	ML	0.1293	0.391E-14	0.133E-14	0.524E-14	0.316E+01
	TH	0.0250	0.169E+02	0.578E+01	0.226E+02	0.894E+01
	OU	0.0067	0.128E-14	0.231E-05	0.231E-05	0.707E+01
M=50	GS	0.0027	0.220E-14	0.542E-04	0.542E-04	0.707E+01
	TL	0.0047	0.724E-02	0.749E-02	0.147E-01	0.708E+01
Cond(C)	QR	0.1269	0.827E-14	0.272E-02	0.272E-02	0.707E+01
=.594E15	ML	3.6858	0.330E-13	0.804E-04	0.804E-04	0.707E+01
	TH	1.8237	0.415E+02	0.139E+02	0.554E+02	0.210E+02
	OU	0.0110	0.255E-14	0.666E-14	0.921E-14	0.100E+02
M=100	GS	0.0053	0.374E-14	0.711E-14	0.108E-13	0.100E+02
	TL	0.0097	0.982E-02	0.127E-02	0.111E-01	0.100E+02
Cond(C)	QR	0.8596	0.114E-13	0.866E-14	0.201E-13	0.100E+02
=.167E03	ML	18.3306	0.875E-13	0.326E-13	0.120E-12	0.100E+02
	TH	13.9822	0.594E+02	0.198E+02	0.792E+02	0.298E+02
	OU	0.0258	0.293E-14	0.602E-04	0.602E-04	0.141E+02
M=200	GS	0.0103	0.461E-14	0.215E-03	0.215E-03	0.141E+02
	TL	0.0164	0.705E-02	0.158E-01	0.228E-01	0.142E+02
Cond(C)	QR	6.9812	0.322E-13	0.223E-03	0.223E-03	0.141E+02
=.594E15	ML		0.239E-12	0.474E-04	0.474E-04	0.141E+02
	TH					
	OU	0.0325	0.117E-13	0.675E-13	0.792E-13	0.173E+02
M=300	GS	0.0151	0.626E-13	0.746E-13	0.809E-13	0.173E+02
	TL	0.0267	0.119E-01	0.249E-02	0.144E-01	0.173E+02
Cond(C)	QR	22.3780	0.417E-13	0.497E-13	0.914E-13	0.173E+02
=.498E03	ML		0.405E-12	0.171E-12	0.575E-12	0.173E+02
	TH					
	OU	0.0435	0.515E-14	0.124E-12	0.129E-12	0.200E+02
M=400	GS	0.0200	0.734E-14	0.139E-12	0.146E-12	0.200E+02
	TL	0.0315	0.115E-01	0.285E-02	0.143E-01	0.200E+02
Cond(C)	QR	50.7211	0.593E-13	0.128E-12	0.187E-12	0.200E+02
=.663E03	ML		0.598E-12	0.277E-12	0.875E-12	0.200E+02
	TH					
	OU	0.0776	0.456E-14	0.569E-04	0.569E-04	0.224E+02
M=500	GS	0.0241	0.749E-14	0.571E-03	0.571E-03	0.224E+02
	TL	0.0370	0.704E-02	0.250E-01	0.320E-01	0.224E+02
Cond(C)	QR	99.4693	0.957E-13	0.659E-04	0.659E-04	0.224E+02
=.637E15	ML		0.832E-12	0.876E-03	0.876E-03	0.224E+02
	TH					

Table 5(Results of computations of system 5 with 5 identical right-hand sides)

	No pr.	Com.time (sec.)	D1= $\|Y - C\tilde{X}\|$	D2= $\|\|X\| - \|\tilde{X}\|\|$	D2+D1	$\|\tilde{X}\|$
	OU	0.0021	0.000E+00	0.000E+00	0.000E+00	0.316E+01
M=10	GS	0.0015	0.831E-15	0.000E+00	0.831E-15	0.316E+01
	TL	0.0027	0.467E+00	0.312E-01	0.498E+00	0.319E+01
Cond(C)	QR	0.0185	0.596E-14	0.888E-15	0.685E-14	0.316E+01
=.107E03	ML	0.6685	0.178E-13	0.155E-14	0.194E-13	0.316E+01
	TH	0.1245	0.452E+03	0.360E+02	0.488E+03	0.392E+02
	OU	0.0056	0.000E+00	0.000E+00	0.000E+00	0.707E+01
M=50	GS	0.0057	0.179E-14	0.688E-14	0.867E-14	0.707E+01
	TL	0.0166	0.845E+00	0.312E-01	0.877E+00	0.710E+01
Cond(C)	QR	0.6300	0.431E-13	0.202E-13	0.633E-13	0.707E+01
=.186E05	ML	17.9965	0.115E-12	0.139E-12	0.254E-12	0.707E+01
	TH	9.0150	0.109E+04	0.840E+02	0.117E+04	0.911E+02
	OU	0.0110	0.000E+00	0.000E+00	0.000E+00	0.100E+02
M=100	GS	0.0112	0.297E-14	0.582E-11	0.582E-11	0.100E+02
	TL	0.0339	0.859E+00	0.249E-01	0.884E+00	0.100E+02
Cond(C)	QR	4.1670	0.707E-13	0.242E-09	0.242E-09	0.100E+02
=.494E08	ML	93.5945	0.309E-12	0.950E-10	0.953E-10	0.100E+02
	TH	60.3865	0.155E+04	0.119E+03	0.167E+04	0.129E+03
	OU	0.0245	0.000E+00	0.000E+00	0.000E+00	0.141E+02
M=200	GS	0.0221	0.445E-14	0.363E-05	0.363E-05	0.141E+02
	TL	0.0589	0.869E+00	0.194E-01	0.888E+00	0.142E+02
Cond(C)	QR	34.7995	0.167E-12	0.196E-03	0.196E-03	0.141E+02
=.437E14	ML		0.817E-12	0.181E-03	0.181E-03	0.141E+02
	TH					
	OU	0.0364	0.000E+00	0.000E+00	0.000E+00	0.173E+02
M=300	GS	0.0343	0.338E-12	0.171E+04	0.171E+04	0.173E+04
	TL	0.0872	0.306E+02	0.722E+04	0.725E+04	0.724E+04
Cond(C)	QR		0.117E-11	0.404E+03	0.404E+03	0.421E+03
=.000E00	ML		0.269E+04	0.207E+03	0.290E+04	0.225E+03
	TH					
	OU	0.0541	0.000E+00	0.000E+00	0.000E+00	0.200E+02
M=400	GS	0.0545	0.904E-06	0.305E+10	0.305E+10	0.305E+10
	TL	0.1305	0.443E+02	0.128E+11	0.128E+11	0.128E+11
Cond(C)	QR		0.457E-12	0.717E+02	0.717E+02	0.917E+02
=.000E00	ML		0.311E+04	0.240E+03	0.335E+04	0.260E+03
	TH					
	OU	0.0693	0.000E+00	0.000E+00	0.000E+00	0.224E+02
M=500	GS	0.0686	0.362E+00	0.269E+16	0.269E+16	0.269E+16
	TL	0.1474	0.566E+02	0.113E+17	0.113E+17	0.113E+17
Cond(C)	QR		0.533E-11	0.248E+04	0.248E+04	0.251E+04
=.000E00	ML		0.348E+04	0.268E+03	0.375E+04	0.290E+03
	TH					

Table 6(Results of computations of system 5 with 5 perturbed (≈ 1%) identical right-hand sides)

	No pr.	Com.time (sec.)	D1= $\|Y - C\tilde{X}\|$	D2= $\|\|X\| - \|\tilde{X}\|\|$	D2+D1	$\|\tilde{X}\|$
	OU	0.0023	0.236E-14	0.316E-01	0.316E-01	0.319E01
M=10	GS	0.0018	0.135E-14	0.316E-01	0.316E-01	0.319E01
	TL	0.0028	0.317E00	0.495E-01	0.367E00	0.321E01
Cond(C)	QR	0.0180	0.984E-14	0.316E-01	0.316E-01	0.319E01
=.107E03	ML	0.6799	0.236E-13	0.316E-01	0.316E-01	0.319E01
	TH	0.1398	0.452E03	0.360E02	0.488E03	0.392E02
	OU	0.0059	0.491E-14	0.707E-01	0.707E-01	0.714E01
M=50	GS	0.0056	0.305E-14	0.707E-01	0.707E-01	0.714E01
	TL	0.0155	0.145E01	0.155E00	0.160E01	0.723E01
Cond(C)	QR	0.6300	0.437E-13	0.707E-01	0.707E-01	0.714E01
=.186E05	ML	18.6911	0.167E-12	0.707E-01	0.707E-01	0.714E01
	TH	8.7856	0.108E04	0.840E02	0.117E04	0.911E02
	OU	0.0113	0.712E-14	0.100E00	0.100E00	0.101E02
M=100	GS	0.0112	0.407E-14	0.100E00	0.100E00	0.101E02
	TL	0.0455	0.145E01	0.162E00	0.162E01	0.102E02
Cond(C)	QR	4.3405	0.801E-13	0.100E00	0.100E00	0.101E02
=.494E08	ML	92.5906	0.430E-12	0.100E00	0.100E00	0.101E02
	TH	65.1850	0.155E04	0.119E03	0.167E04	0.129E03
	OU	0.0250	0.996E-14	0.141E00	0.141E00	0.143E02
M=200	GS	0.0230	0.545E-14	0.141E00	0.141E00	0.143E02
	TL	0.0610	0.147E01	0.188E00	0.165E01	0.143E02
Cond(C)	QR	33.1890	0.175E-12	0.141E00	0.141E00	0.143E02
=.437E14	ML		0.105E-11	0.143E00	0.143E00	0.143E02
	TH					
	OU	0.0551	0.368E-13	0.173E00	0.173E00	0.175E02
M=300	GS	0.0553	0.118E-11	0.238E04	0.238E04	0.240E04
	TL	0.1130	0.521E02	0.359E04	0.364E04	0.361E04
Cond(C)	QR		0.112E-11	0.321E03	0.321E03	0.338E03
=.000E00	ML		0.272E04	0.210E03	0.293E04	0.227E03
	TH					
	OU	0.0545	0.142E-13	0.200E00	0.200E00	0.202E02
M=400	GS	0.0549	0.169E-05	0.424E10	0.424E10	0.424E10
	TL	0.1545	0.813E02	0.634E10	0.634E10	0.634E10
Cond(C)	QR		0.479E-12	0.762E02	0.762E02	0.962E02
=.000E00	ML		0.314E04	0.242E03	0.339E04	0.262E03
	TH					
	OU	0.0698	0.158E-13	0.224E00	0.224E00	0.226E02
M=500	GS	0.0694	0.162E01	0.374E16	0.374E16	0.374E16
	TL	0.1850	0.813E02	0.560E16	0.560E16	0.560E16
Cond(C)	QR		0.506E-11	0.251E04	0.251E04	0.253E04
=.000E00	ML		0.352E04	0.271E03	0.379E04	0.293E03
	TH					

The general analysis of the results of testing (in particular, presented in Tables 3 ÷ 6) shows that our program OU (DCSOL1) has better accuracy characteristics but is about twice as worse in time as the program GS (DBEQN) . This is due to the time consumption on the analysis of C and $B = C^{-1}$ in accordance with our algorithm. However, if the system $CX = Y$ is solved with several right–hand sides, the indicated analysis is performed only once. In this case our program DCSOL1 does not concede the program GS (DBEQN) also in time, starting from 2 right-hand sides. This is confirmed by analysing Tables 5, 6, where we reported the results of solution of the system $CX = Y$ with 5 identical right–hand sides.

In Table 6, we place the results of numerical computations on the solution of the perturbed ($\approx 1\%$ in Y) system with 5 identical right–hand sides. As it follows from the analysis of the results of testing and, in particular, Table 6, our method is the most stable also to the perturbation of the right–hand side of the system of equations.

6. Conclusion

We have developed the effective method and algorithm of inversion of tridiagonal matrices of the general form and also the algorithm of solution of systems of algebraic equations with those matrices. The programs created on their basis surpass in basic characteristics all known modern programs of a similar type.

References

[1] Emel'yanenko G.A., Rakhmonov T.T., Dushanov E.B. Critical - component method for solving systems of linear equations with a tridiagonal matrix of the general form. JINR preprint, E11-96-105, Dubna, 1996.

[2] Emel'yanenko G.A., Rakhmonov T.T. Generators of matrix– factorized representations of block–tridiagonal matrices and their inverses. JINR preprint, P11-93-265, Dubna, 1993.

[3] Emel'yanenko G.A. Rakhmonov T.T. Decomposition of the region and generators of direct methods of systems of linear equations with block–tridiagonal matrices of the general form. JINR preprint, P11-93-266, Dubna, 1993.

[4] Emel'yanenko G.A. Rakhmonov T.T. On structure of direct representations of matrices inverse of block–tridiagonal ones with vanishing leading block angular minors. JINR preprint, P11-93-248, Dubna, 1993.

[5] Malyshev A.N. Introduction to computational linear algebra (with application of algorithms on FORTRAN). Novosibirsk, "Nauka", 1991.

[6] Faddeeva V.N., Kolotilina L.Yu. Computational methods of linear algebra: set of matrices for testing, p.1, p.2, p.3 (Materials on mathematical software of IBM), Leningrad, 1982.

[7] Molchanov I.N. Computer methods of solving applied problems. Differential equations. Kiev, "Nauk.dumka", 1988.

[8] CERNLIB – CERN Program Library (Short Writeups). Application Software Group. Computing and Networks Division. CERN, Geneva, Switzerland (May, 1993).

[9] Rice J. Matrix computations and mathematical software. McGraw–Hill Book Company, 1981.

[10] Fedorova R.N., Shirokova A.I. Library of programs on FORTRAN. vv.$VI - VII$. Description of programs. Dubna, 1990.

[11] Bogolubov A.N., Telegin V.I. On one numerical method of solution of systems of linear equations with a tridiagonal matrix. JCM and MPh, 1974, 14, No 3, c.768-771.

Received by Publishing Department on March 28, 1996.