Engineering Simulation, 1995, Vol. 13, pp. 259-270
Reprints available from the Publisher
Photocopying permitted by license only

Processor Arrays for Solving Nonstationary Systems of Equations of Mathematical Physics

N. A. LIKHODED, A. A. TIUNCHIK, and V.A. TSURKO
Institute of Mathematics, National Academy of Sciences of Belorus, Minsk

(Received December 10, 1993)

In this paper systolic processors are suggested for applying the Gauss block method to solve systems of linear algebraic equations appearing in numerical solution of nonstationary equations of mathematical physics. These processors are characterized by much weaker hardware requirements for the same computation time as compared to the existing structures implementing the Gauss method.

Description of complex physical processes with multicomponent interaction often reduces to nonstationary systems of second-order partial differential equations, e.g., parabolic or hyperbolic equations. As a rule, solution of these problems by analytical methods proves impossible. In this case the main instrument for finding solutions is numerical methods such as finite-difference and finite-element methods whose application requires the solution of a great number of systems of linear algebraic equations (SLAE), which sometimes are of high dimension. In this connection there arises a problem of fast solution of a set of SLAE. Its solution seems to lie in the application of special-purpose high-performance multiprosessor systems, in particular, systolic processors (SP).

We will synthesize an SP that implements the matrix factorization method (the Gauss block method) for solving SLAE of the form

$$\begin{bmatrix} C_0 & B_0 & 0 & 0 & \dots & 0 \\ A_1 & C_1 & B_1 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \dots & A_{N-1} & C_{N-1} & B_{N-1} & \\ 0 & \dots & 0 & A_N & C_N & \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_1 \\ \cdot\cdot \\ Y_{N-1} \\ Y_N \end{bmatrix} = \begin{bmatrix} F_0 \\ F_1 \\ \dots \\ F_{N-1} \\ F_N \end{bmatrix}, \tag{1}$$

where $A_1,\dots, A_N$, $B_0,\dots, B_{N-1}$, $C_0,\dots, C_N$ are dense $M \times M$ matrices and $Y_0,\dots, Y_N$, $F_0,\dots, F_N$ are M-dimensional column vectors. These SLAEs often appear in solving systems of partial differential equations. For example, consider a system of one-dimensional linear parabolic equations in a region $G = [0, l] \times [0, T]$:

$$\frac{\partial U}{\partial t} = \frac{\partial}{\partial x}\left[K\frac{\partial U}{\partial t}\right], \tag{2}$$

$$U(x,0) = U_*, \quad U(0,t) = \mu_1(t), \quad U(l,t) = \mu(t), \tag{3}$$

where $U = U(x, t) = (U^1,\ldots, U^M)$ and $U_* = u_*(x) = (U_*^1,\ldots, U_*^M)$ are M-dimensional vectors and $K = (K_{\alpha\beta}) = (K_{\alpha\beta}(x, t))$, $\alpha, \beta = 1, 2,\ldots, M$, is a positive definite $M \times M$ matrix ($K > 0$ for $(x, t) \in G$). It is assumed that problem (2), (3) has a unique solution $U(x, t)$, which is continuous in G and differentiable as many times as is required for approximation.

Let us introduce the nets $\overline{\omega}_h = \{x_i = ih, i = 0, 1,\ldots, N\}$ and $\overline{\omega}_\tau = \{t_j = j\tau, j = 0, 1,\ldots, j_0\}$ and the net $\overline{\omega}_{h\tau} = \overline{\omega}_h \times \overline{\omega}_\tau = \{(ih, j\tau), i = 0, 1,\ldots, N; j = 0, 1,\ldots, j_0\}$ in G with steps $h = l/N$ and $\tau = T/j$. Denote by y_i^j the approximate value of the vector U at the node (x_i, t_j); let k_i^j be the matrix of the values of the function $K_{\alpha\beta}$ at the same node; $a_i^j = 0.5(k_{i-1}^j + k_i^j)$. Consider the following purely implicit finite-difference scheme approximating equation (2):

$$\frac{y_i^{j+1} - y_i^j}{\tau} = \frac{1}{h^2}\left(a_{i+1}^{j+1}(y_{i+1}^{j+1} - y_i^{j+1}) - a_i^{j+1}(y_i^{j+1} - y_{i-1}^{j+1})\right), \quad 1 \le i \le N-1, \quad 0 \le j \le j_0 - 1. \tag{4}$$

The scheme (4) provides approximation of the order of $O(h^2+\tau)$ on the solution $U = U(x, t)$ and is unconditionally stable and monotone, and the approximate solution converges to the exact one at a rate $O(h^2+\tau)$. The boundary and initial conditions are approximated exactly:

$$y_0^{j+1} = U_0^{j+1}, \quad y_N^{j+1} = U_N^{j+1},$$

$$y_i^0 = U(x_i, 0) = U_*(x_i). \tag{5}$$

For each $j = 0, 1,\ldots, j_0-1$ the solution of problem (5), (6) reduces to solving the SLAE

$$A_i y_{i-1} + C_i y_i + B_i y_{i+1} = F_i, \quad 1 \le i \le N-1,$$

$$y_0 = U_0, \quad y_N = U_N,$$

which can be written in the form (1); for the boundary conditions of the first kind under consideration B_0 and A_N are nonzero $M \times M$ matrices.

Remark. Besides finite-difference schemes (4), (5) with advance we can also apply general schemes with weight λ [1]. For $\lambda \neq 0$ the solution algorithm for the resulting finite-difference schemes practically coincide with those above.

Matrix factorization method for solving SLAE with blocked tridiagonal matrix. The matrix factorization method (the Gauss block method) for solving systems of equations (1)

consists of direct and reverse runs. The direct run reduces to transforming equations (1) to an upper blocked two-diagonal system of equations:

$$\begin{bmatrix} I & U_0 & 0 & \dots & 0 \\ 0 & I & U_1 & \dots & 0 \\ . & . & . & . & . \\ 0 & 0 & \dots & I & U_{N-1} \\ 0 & 0 & 0 & \dots & I \end{bmatrix} \begin{bmatrix} Y_0 \\ Y_1 \\ \dots \\ Y_{N-1} \\ Y_N \end{bmatrix} = \begin{bmatrix} G_0 \\ G_1 \\ \dots \\ G_{N-1} \\ G_N \end{bmatrix}, \tag{6}$$

where I is the unit $M \times M$ matrix,

$$U_0 = C_0^{-1} B_0, \quad G_0 = C_0^{-1} F_0,$$

$$U_q = (C_q - A_q U_{q-1})^{-1} B_q, \quad q = 1,2,\dots,N-1,$$

$$G_q = (C_q - A_q U_{q-1})^{-1} F_q A_q G_{q-1}, \quad q = 1,2,\dots,N. \tag{7}$$

The reverse run is the solution of the system of equations (6):

$$Y_N = G_N, \quad Y_q = G_q - U_q Y_{q+1}, \quad q = N-1, N-2, \dots, 1, 0. \tag{8}$$

Thus, the matrix factorization method for SLAE (1) can be implemented on SPs capable of performing matrix operations of the form

$$(\tilde{C} | \tilde{F}) = (C|F) - A(U|G) \tag{9}$$

(for simultaneous performance of the operations $C_q - A_q U_{q-1}$ and $F_q - A_q G_{q-1}$),

$$(U|G) = \tilde{C}^{-1}(B|\tilde{F}) \tag{10}$$

(for simultaneous performance of the operations $\tilde{C}_q^{-1} B_q$ and $\tilde{C}_q^{-1} \tilde{F}_q$ or $C_0^{-1} B_0$ and $C_0^{-1} F_0$, where the matrix $\tilde{C}_q = C_q - A_q U_{q-1}$ and the column vector $\tilde{F}_q = F_q - A_q G_{q-1}$ result from the operation (9)), and

$$D = G - UY \tag{11}$$

(for performing the reverse run (8); $\tilde{C}$, C, A, U, and B in (9)-(11) are $M \times M$ matrices and $\tilde{F}$, F, G, D, and Y are M-dimensional column vectors.

Systolic processors for performing operations (9)-(11). We will synthesize an SP for performing the operations (9)-(11) using the well-known techniques in [2], [3], [4], [5], [6]. The result of the operation $(U|G) = \tilde{C}^{-1}(B|\tilde{F})$ can be obtained as a solution of the SLAE $\tilde{C}(U|G) = (B|\tilde{F})$ of M equations with M unknowns and $M+1$ right-hand sides. This system of equations can be solved by the Gauss-Jordan method written in the following form:

$$z(i,j,0) = \tilde{c}_{ij}, \quad 1 \le i,j \le M,$$

$$z(i,j+M,0)=b_{ij},\quad 1\le i,j\le M,$$

$$z(i,2M+1,0)=f_i,\quad 1\le i,j\le M;$$

$$1\le k\le M;$$

$$1.\ y(k,k,k)=1/z(k,k,k-1);$$

$$k+1\le i\le K+M-1;$$

$$2.\ y(i,k,k)=-z(i,k,k-1);$$

$$k+1\le j\le 2M+1;$$

$$3.\ x(k,j,k)=z(k,j,k-1)y(k,j-1,k),$$

$$y(k,j,k)=y(k,j-1,k);$$

$$k+1\le i\le K+M-1,\quad k+1\le j\le 2M+1;$$

$$4.\ z(i,j,k)=z(i,j,k-1)+x(i-1,j,k)y(i,j-1,k),$$

$$x(i,j,k)=x(i-1,j,k),$$

$$y(i,j,k)=y(i,j-1,k),$$

$$k+1\le j2M+1;$$

$$5.\ z(k+M,j,k)=x(k+M-1,j,k);$$

$$u_{ij}=z(i+M,j+M,M),\quad 1\le i,j\le M, \tag{12}$$

The graph of the algorithm (12) for $M = 3$ is shown in Figure 1, *a*, where the vertex types are marked by the corresponding numbers. To the five types of elementary operations in the algorithm (12) there correspond five types of graph vertices.

The graph vertices are located in the region $V = \{u(i, j, k) \in Z^3 \mid 1 \le k < M, k \le$

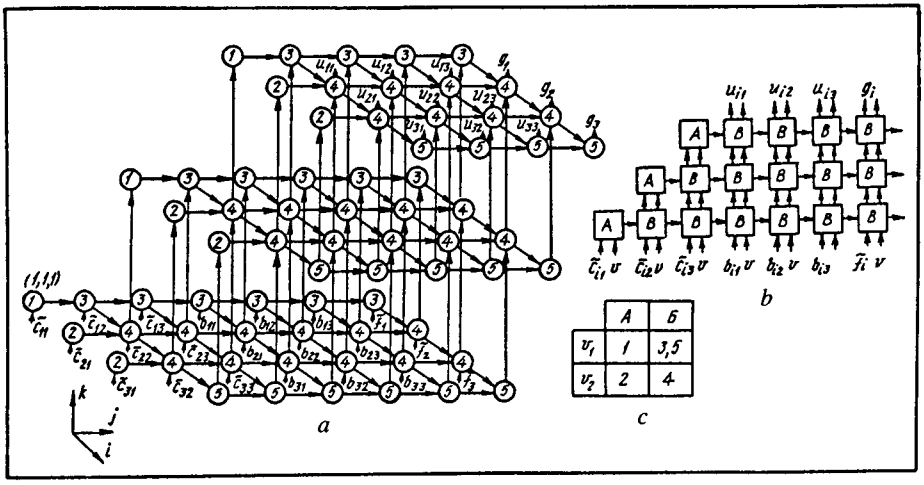

Figure 1

$i \le k+M$, $k \le j \le 2M+1$; $j \ne k$ for $i = k+M\}$, and the arcs are characterized by the vectors (1, 0, 0), (0, 1, 0), and (0, 0, 1).

The resulting graph can be mapped into the two-dimensional systolic processor SP_{LS} (Figure 1, *b*) using the allocation function

$$f_{LS}(v(i,j,k)) = (j,k) \tag{13}$$

and the timing function

$$t_{LS}(v(i,j,k)) = \tau_{LS} + (i-1) + (j-1) + (k-1), \tag{14}$$

where τ_{LS} is a constant determining the time instant when SP_{LS} starts its operation.

The application of the allocation function (13) results in the mapping of type 1 and 2 vertices and type 3-5 vertices of the graph into one (type A) and one (type B) processor elements (PE), respectively. Operating modes for these PEs are controlled by the signals ν_1 and ν_2 that are fed to the graph vertices $v(1, j, 1) \in V$, $1 \le j \le 2M+1$, and $v(i, j, 1) \in V$, $2 \le i \le M$, $1 \le j \le 2M+1$, and then transported in the direction of the vector (1, 0, 1), The correspondence between the types of PEs, types of graph vertices, and control signals is demonstrated in Figure 1, *c*.

To perform the operation $(C|\tilde{F}) = (C|F) - A(U|G)$ we employ the following algorithm:

$$p(i,0,k) = -a_{ik}, \quad 1 \le i,k \le M,$$

$$z(0,j,k) = u_j, \quad 1 \le i,j \le M,$$

$$z(0,M+1,k) = g, \quad 1 \le k \le M,$$

$$q(i,j,M+1) = c_{ij}, \quad 1 \le i,j \le M,$$

$$q(i,M+1,M+1) = f_i, \quad 1 \le i \le M;$$

$$M \ge k \ge 1, \quad 1 \le i \le M, \quad 1 \le j \le M+1;$$

$$6.\ p(i,j,k) = p(i,j-1,k),$$

$$z(i,j,k) = z(i-1,j,k),$$

$$q(i,j,k) = p(i,j,k+1) + p(i,j-1,k) \cdot z(i-1,j,k);$$

$$\tilde{c}_{ij} = q(i,j,1), \quad 1 \le i,j \le M,$$

$$\tilde{f}_i = q(i,M+1,1), \quad 1 \le i \le M. \tag{15}$$

The vertices of the graph for algorithm (15) are located in the region $V = \{v(i, j, k) \in Z^3 \mid 1 \le i \le M, 1 \le j \le M+1, 1 \le k \le M\}$ and the arcs are characterized by the vectors (1, 0, 0), (0, 1, 0), and (0, 0, −1). The resulting graph can be mapped into the SP by means of the allocation function

$$f_{MM}(v(i,j,k)) = (j+k,k). \tag{16}$$

Under this mapping the elements of the matrix $(U|G)$ must be loaded into the local memory of the PE of the synthesized SP. To overlap the loading of the elements of the matrix $(U|G)$ with calculations we complete the region V with the vertices $\{v(i, j, k) \in Z^3 \mid 1 \le j \le M+1, 2 \le k \le M, 2-k \le i \le 0\}$ and connect them with arcs in the direction of the vector (1, 0, −1) as is shown in Figure 2, *a*.

The space-time mapping of the extended graph by means of the allocation function (16) and timing function

$$f_{MM}(v(i,j,k) = \tau_{MM} + (i+M-2) + (j-1) - (k-M), \tag{17}$$

where τ_{MM} is a constant, results in the systolic processor SP_{MM} shown in Figure 2, *b*. The processor elements of SP_{MM} operate in two regimes (Figure 2, *c*).

Let us synthesize an SP for performing the operation $D = G - UY$. The calculation algorithm for D can be written in the following form:

$$g(i,0) = g_i, \quad 1 \le i \le M,$$

$$y(0,k) = y_k, \quad 1 \le k \le M;$$

$$1 \le k \le M, \quad 1 \le i \le M:$$

$$8.\ y(i,k) = y(i-1,k),$$

$$g(i,k) = g(i,k-1) - u_{ik} \cdot y(i-1,k);$$

$$d_i = g(i,M), \quad 1 \le i \le M. \tag{18}$$

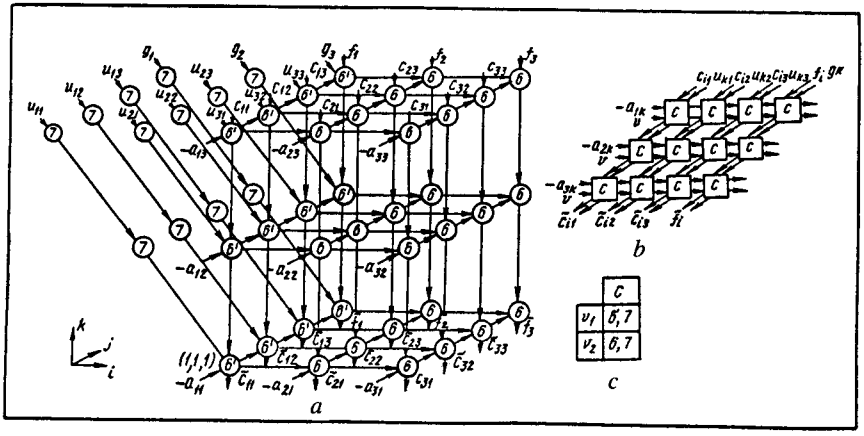

Figure 2

If the graph for algorithm (18) is extended by adding type 9 vertices (for transporting y_k) as is shown in Figure 3, *a* and then is mapped on a linear SP by means of the allocation function $f_{MV}(v(i,\ k)) = k$ and timing function

$$f_{MV}(v(i,k)) = \tau_{MV} + 2(i + M - 2) + (k - M), \tag{19}$$

this results in the systolic processor SP_{MM} represented in Figure 3, *b*.

Systolic device for performing the direct run. The direct run of the matrix factorization method with global interconnections can be performed on a systolic device for solving SLAE of the form (1), which consist of SP_{LS} and SP_{MM} (Figure 4, *a*). Indeed, according

to algorithm (7), SP_{LS} permits calculating $(U_0|G_0) = C_0^{-1}(B_0|F_0)$ and $(U_q|G_q) = \tilde{C}_q^{-1}(B_q|F_q)$, $1 \le q \le N$, where B_N is an arbitrary $M \times M$ matrix, and SP_{MM} can be used for calculating $(\tilde{C}_q|\tilde{F}_q) = (C_q|F_q) - A_q(U_{q-1}|G_{q-1})$, $1 \le q \le N$.

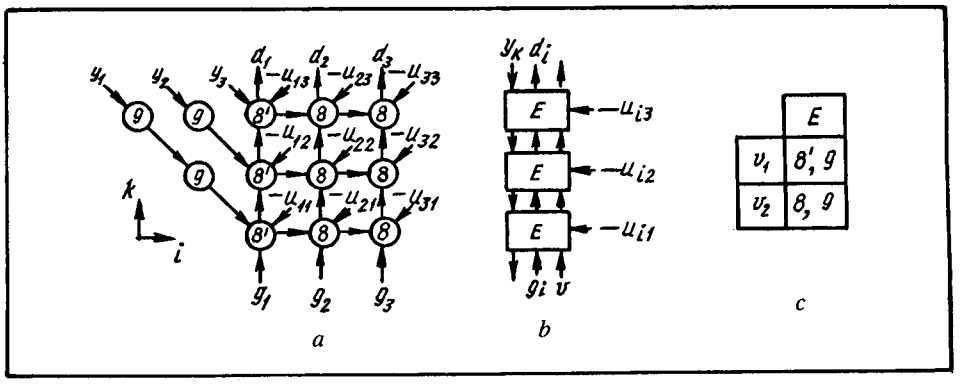

Figure 3

We will show that the calculations on SP_{LS} and SP_{MM} can be carried out in parallel. Let $t_{LS}^{(q)}(v)$ denote a timing function of the form (14) in the calculation of the matrix $(U_q|G_q)$ on the SP_{LS} ($0 \le q \le N$) and let $t_{MM}^{(q)}(v)$ be a timing function of the form (17) in the calculation of the matrix $(\tilde{C}_q|\tilde{F}_q)$ on the SP_{MM} ($1 \le q \le N$). The systolic processors SP_{LS} and SP_{MM} are data-dependent by means of the matrices $(U_q|G_q)$ and $(\tilde{C}_q|\tilde{F}_q)$. According to the timing functions (14) and (17), the elements $u_{ij}^{(q)}$, $1 \le i, j \le M$, of the matrix U_q issue from SP_{LS} at times $t_{LS}^{(q)}(v(i+M, j+M, M))+1 = t_{LS}^{(q)}+i+j+3M-2$ and arrive at SP_{MM} at times $t_{MM}^{(q+1)}(v(i-M+1, j, M)) = \tau_{MM}^{(q+1)}+i+j-2$ and the elements $g_i^{(q)}$, $1 \le i \le M$, of the vector G_q issue from SP_{LS} at times $t_{LS}^{(q)}(v(i+M, 2M+1, M))+1 = \tau_{LS}^{(q)}+i+4M-1$ and arrive at SP_{MM} at times $t_{MM}^{(q+1)}(v(i-M+1, M+1, M))+1 = \tau_{MM}^{(q+1)}+i+M-1$. Consequently, these elements can be transferred from SP_{LS} to SP_{MM} immediately if we choose

$$\tau_{MM}^{(q+1)} = \tau_{LS}^{(q)} + 3M. \tag{20}$$

Because, according to the properties of the timing functions (14) and (17), the elements $\tilde{c}_{ij}^{(q)}$, $1 \le i, j \le M$, of the matrix $\tilde{C}_q$ issue from SP_{MM} at times $\tau_{MM}^{(q)}(v(i, j, 1))+1 = \tau_{MM}^{(q)}+i+j+2M-3$ and arrive at SP_{LS} at times $\tau_{LS}^{(q)}(v(i, j, 1)) = \tau_{LS}^{(q)}+i+j-2$, they can be transferred from SP_{MM} to SP_{LS} immediately if we choose

$$\tau_{LS}^{(q)} = \tau_{MM}^{(q)} + 2M - 1. \tag{21}$$

Formulas (20) and (21) determine an optimal choice of the parameters $\tau_{LS}^{(q)}$, $0 \le q \le N$, and $\tau_{MM}^{(q)}$, $1 \le q \le N$, for the solution of one equation (1): $\tau_{LS}^{(q)} = (5M-1)q$ and $\tau_{MM}^{(q)} = (5M-1)(q-1)+3M$.

When not one but many equations of the form (1) are solved, the chosen parameters

$\tau_{LS}^{(q)}$ and $\tau_{MM}^{(q)}$ should be proportional to the SP_{LS} and SP_{MM} calculation time, which is equal to M (with each step q of algorithm (7) increased by one clock cycle using unit delays D): $\tau_{LS}^{(q)} = 5Mq$, $0 \le q \le N$, and $\tau_{MM}^{(q)} = 5M(q-1)+3M$, $1 \le q \le N$. The choice of these $\tau_{LS}^{(q)}$ and $\tau_{MM}^{(q)}$ makes it possible to solve five problems simultaneously.

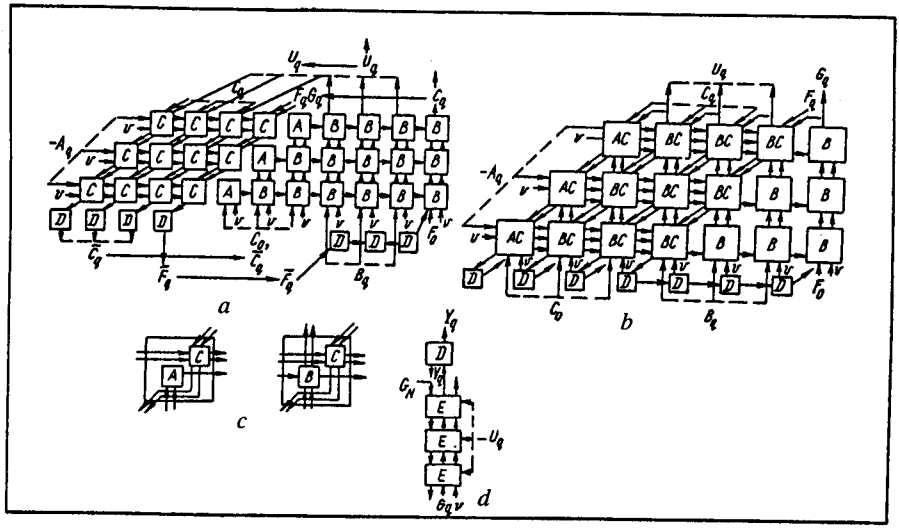

Figure 4

The elements $\tilde{f}_i^{(q)}$, $1 \le i \le M$, of the vector $\tilde{F}_q$ are taken from SP_{MM} at times $t_{MM}^{(q)}(v(i, M+1, 1))+1 = \tau_{MM}^{(q)}+i+3M-2$ and then fed to SP_{LS} at times $t_{LS}^{(q)}(v(i, 2M+1, 1)) = \tau_{LS}^{(q)}+i+2M-1$. Because $\tau_{LS}^{(q)}+i+2M-1-(\tau_{MM}^{(q)}+i+3M-2) = M+1$, the element $\tilde{f}_i^{(q)}$ should be delayed by $M+1$ time cycles after it issues from SP_{MM}.

The synthesized systolic device contains $(5/2)(M^2+M)$ processor elements performing operations of division or multiplication with addition. The calculation time for the direct run for one problem is $t_{LS}^{(N)}(v(2M, 2M+1, M)-t_{LS}^{(0)}(v(1, 1, 1))+1 = 5MN+5M-1$, and for five problems it is $t_{LS}^{(N)}(v(2M, 2M+1, M)-t_{LS}^{(0)}(v(1, 1, 1))+1+4M = 5MN+9M-1$. Note that the existing SPs [7], [8], [9] implementing the direct run in the Gauss method for solving SLAEs with band matrices (including SLAEs of the form (1)) contains $4M^2+O(M)$ processor elements; in this case the performance time for Λ direct runs is approximately equal to ΛMN as in the case of the synthesized SP.

A systolic processor for performing the direct run without global interconnections can be obtained by "superposition" of SP_{LS} and SP_{MM} (Figure 4, *b*). Each PE of this SP includes processor elements SP_{LS} and SP_{MM} that can operate independently. In this case the time characteristics and the number of arithmetic devices are the same as in the case of the systolic device represented in Figure 4, *a* but the global interconnections between

them are eliminated.

Systolic device for performing the reverse run. The reverse run (8) is implemented by SP_{MV} for multiplying a matrix by a vector and one register (Figure 4, *e*). The vectors G_N, $G_{N-1},\ldots, G_0, -U_N, U_{N-1},\ldots, -U_0$ are transferred to SP_{MV} from the external memory. The calculated vectors Y_{q+1}, $0 \le q < N-2$, are returned back to the SP_{MV} for calculating Y_q.

We will prove the compatibility of the steps in the reverse run on the specialized systolic processor. Denote by $t_{MV}^{(q)}(v)$ the timing function of the form (19) for calculating the vector Y_q. The elements $y_i^{(q+1)}$, $1 \le i \le M$, of the vector Y_{q+1} issue from the register at times $t_{MV}^{(q+1)}(v(i, M))+1+1 = \tau_{MV}^{(q+1)}+2i+2M-2$ (see Figures 3, *a* and 4, *e*) and are transferred back to the SP_{MV} at times $t_{MV}^{(q)}(v(i-M+1, M)) = \tau_{MV}^{(q)}+2i-2$. Consequently, the calculated elements $y_i^{(q+1)}$ can be fed to the device immediately if we choose $\tau_{MV}^{(q)} = \tau_{MV}^{(q+1)}+2M = 2(N-1-q)M$, $0 \le q \le N-1$.

The total computing time for realizing the reverse run on the specialized systolic processor is

$$t_{MV}^{(0)}(v(M,M)) - \tau_{MV}^{(N+1)}(v(2-M,M)) + 2 = 2NM = 2M-2.$$

We note that the PEs of the constructed SP perform useful calculations at every other time cycle. Consequently, the number of PEs can be reduced two-fold or two problems can be solved simultaneously without increasing the required time. The well-known SP [7] implementing the solution of SLAE with band matrices has twice as many processor elements as the synthesized SP; the calculation time for the reverse run is the same for the two SPs.

On some problems solved using the synthesized SPs. The synthesized SPs can be applied for solving many-dimensional systems of equations of the form

$$\frac{\partial U}{\partial t} = \sum_{\gamma=1}^{P} L_\gamma, \quad 0 \le t \le T \tag{22}$$

in the P-dimensional parallelepiped $G = \{0 < x_\gamma < l_\gamma, \gamma = 1, 2,\ldots, P\}$, where $L_\gamma = \frac{\partial}{\partial x_\gamma}\left[K_\gamma \frac{\partial U}{\partial x_\gamma}\right]$; and $K_\gamma = K_\gamma(x, t)$ are square $M \times M$ matrices, $\gamma = 1, 2,\ldots, P$, $x = x(x_1,\ldots, x_P)$. To this end we use the following technique for constructing locally one-dimensional schemes [1]. We associate with equation (22) a chain P of systems having a simpler structure:

$$\frac{1}{P}\frac{\partial V_\gamma}{\partial t} = L_\gamma, \quad x \in G, \quad t_{j+(\gamma-1)/P} \le t \le t_{j+\gamma/P}, \quad 1 \le \gamma \le P$$

with conditions $V_1(x, 0) = U_0(x)$, $V_\gamma(x, t_{j+(\gamma-1)/P}) = V_{\gamma-1}(x, t_{j+(\gamma-1)/P})$, $1 \le \gamma \le P$, $1 \le j \le j_0$, $V_1(x, t_j) = V_P(x, t_j)$. To solve these systems we can apply the above computational procedures. The value $V_P = U+O(\tau)$ is taken as an approximate solution to problem (22) in each time layer.

Solution of systems of hyperbolic second-order equations

$$\frac{\partial^2 U}{\partial t^2} = \frac{\partial}{\partial X}\left[K\frac{\partial U}{\partial t}\right], \tag{23}$$

(where K is an $M \times M$ matrix) by the finite-difference method also reduces to equations of the form (1). In this case the differential operators in equation (23) are approximated according to the technique in [1].

The synthesized SPs can be used for solving quasilinear and nonlinear systems of equations with general boundary conditions. To this end it is necessary to construct the corresponding iterative process [1] whose every approximation is determined from a system of the form (1). In the case of boundary conditions of the second and third kind the blocks B_0 and A_N are nonzero.

The suggested SPs can be effectively used to solve modeling problems for transistor structures of integrated circuits. The elements of integrated circuits are formed as a result of redistribution of impurities in the process of thermal annealing of semiconductor materials. The diffusion of impurities is described by a system of equations of the type (2) or (22). Investigation of actual technological processes involves matrices of dimension from 2 to 4. Multivariant calculations can be carried out for these problems only using high-performance computers.

Thus, we have synthesized SPs for solving systems of linear algebraic equations with blocked tridiagonal matrix by the Gauss method. The necessity in solving a large number of such SLAEs often arises in the solution of problems of mathematical physics. The suggested SPs require much less (by almost a quarter) hardware for the same computing time as compared to the well-known systolic structures implementing the Gauss method.

REFERENCES

1. A. A. Samarskiy, *Teoriya raznostnykh skhem* (Theory of finite-difference schemes)(Moscow: Nauka, 1989)(in Russian).
2. P. Quinton, in: *Automata Networks in Computer Science.* Ch. 9 (Princeton: Princeton University Press, 1987): 229-260.
3. D. I. Moldovan and J. A. Fortes, *IEEE Trans. Comput.* **35**, No. 1: 1-12(1986).
4. S. Kun, *Matrichnye protsessory na SBIS* (VLSI array processors)(Moscow: Mir, 1991)(Russian translation).
5. Yu. S. Kanevskii, *Sistolicheskie protsessory* (Systolic processors)(Kiev: Tekhnika, 1991)(in Russian).
6. V. Kosianchouk, N. A. Likhoded, and P. I. Sobolevskii, *Systolic architecture array synthesis* (Minsk: Preprint No. 6, Institute of Mathematics, Acad. of Sci. of Belarus, 1992).
7. H. T. Kung and C. E. Leiserson, in: *Sparse Matrix Processing* (Phil.: SIAM. 1979): 256-282.
8. Y. Robert, *Int. Comput. Math.* **17**: 295-315(1985).
9. D. J. Evans and K. Margaritis, *Integration* **8**: 65-90(1989).
10. A. F. Burenkov, A. I. Kirkovskiy, and V. A. Tsurko, *Matem. Modelirovanie* **2**, No. 10: 21-33 (1990).