

PARALLEL ITERATIVE METHODS FOR SOLVING ELLIPTIC EQUATIONS ON TRIANGULAR AND TETRAHEDRAL GRIDS

O. Yu. Milyukova* and I. V. Popov*

**Institute for Mathematical Modeling RAS
125047 Moscow, Russia
125047 Moscow, Russia
Email: miliukova@imamod.ru*

Key words: triangular and tetrahedral grids, parallel iterative methods, ordering of grid points, estimates for the number of iterations

Abstract. Parallel versions of the variant of modified incomplete Cholesky conjugate gradient method are proposed for solving elliptic equations on triangular and tetrahedral grids on MIMD parallel computers. The construction of parallel methods is based on the use special orderings of grid points correlated with domain decomposition. The convergence rates of the proposed parallel methods are examined both theoretically and numerically by analyzing a number of model problems. It is proved and numerically demonstrated that the number of iterations required for convergence of the proposed parallel methods is $O(\sqrt[4]{N})$ for fixed processors number in 2D problems solved on uniform triangular grids and $O(\sqrt[6]{N})$ for fixed processors number in 3D problems solved on nearly uniform tetrahedral grids, where N is the number of grid points. The increase in the number of iterations with the number of processors is acceptable for moderate number of processors. The proposed parallel methods have sufficiently simple algorithm.

1. INTRODUCTION

Elliptic equations are used in mathematical modeling of problems in various fields of physics and engineering, such as fluid dynamics, radiative gas dynamics, oil production, and some problems in microelectronics. Frequently, most of the processing time required to solve a particular problem is expended on the numerical solution of an elliptic boundary value problem. The current rapid progress in high-performance computers with large memory capacity, including multiprocessor systems, makes it possible to use fine unstructured triangular and tetrahedral grids. In turn, this motivates the development of fast converging parallel methods for solving systems of sparse linear algebraic equations with arbitrary coefficient matrices.

In this paper, we consider the symmetric positive definite system

$$Ay = f, \quad A = A^T > 0 \quad (1)$$

as a triangular or tetrahedral discretization of a boundary value problem on an arbitrary simply connected domain for the elliptic equation

$$\sum_{\alpha=1}^m \frac{\partial}{\partial x_{\alpha}} \chi_{\alpha} \frac{\partial u}{\partial x_{\alpha}} = -\varphi, \quad (2)$$

with coefficients $\chi_{\alpha} \geq c_0 > 0$, where $\alpha = 1, \dots, m$ ($m = 2, 3$ is the space dimension). The grid is generated by methods^{1,2}. Equation (2) is discretized by a finite volume method³ or a finite element method. System (1) should be solved by a preconditioned conjugate gradient method, such as ICCG(0) (incomplete Cholesky conjugate gradient, level zero)⁴, the version of MICCG(0) (modified incomplete Cholesky conjugate gradient)⁵ originally proposed in the article⁶ (hereinafter referred to as VMICCG), or the efficient, easily parallelizable methods^{7,8}.

In this paper, we propose a finite volume approximation of 3D equation (2) using piecewise linear interpolating functions. The coefficient matrix of the resulting system of equations is symmetric and positive definite. System (1) is solved here by using VMICCG or VMICCG with regularized preconditioner (on 3D domains). We propose parallel versions of these methods designed to solve equation (1) on MIMD parallel computers, based on orderings of grid points correlated with domain decomposition (Domain Decomposition Ordering, or DDO). The convergence rates of the proposed parallel methods are examined both theoretically and numerically by analyzing several model problems. The proposed parallel methods have sufficiently simple algorithm and are relatively fast converging.

2. DISCRETIZATION METHOD FOR ELLIPTIC BOUNDARY VALUE PROBLEMS

In this section, we propose a method for discretizing 3D elliptic boundary value problems on tetrahedral grids. A detailed analysis of discretization on unstructured triangular grids was presented in³. We consider equation (2) with $\chi_i = \chi$ ($i = 1, 2, 3$). To simplify analysis, we assume that the coefficient χ is constant and the right-hand side of (2) is zero. We rewrite the resulting Laplace equation in conservative form, using x, y, z instead of x_1, x_2, x_3 and U instead of u :

$$\frac{\partial}{\partial x} W^x + \frac{\partial}{\partial y} W^y + \frac{\partial}{\partial z} W^z = 0, \quad (3)$$

$$\vec{W} = \left(\frac{\partial U}{\partial x}, \frac{\partial U}{\partial y}, \frac{\partial U}{\partial z} \right). \quad (4)$$

To discretize this flux equation on an unstructured tetrahedral grid, we approximate $U = U(x, y, z)$ inside each tetrahedron adjacent to a grid vertex as follows:

$$U = U_0 + A(x - x_0) + B(y - y_0) + C(z - z_0),$$

where (x_0, y_0, z_0) are the coordinates of the vertex and U_0 is the corresponding value of the solution. Setting $U = U_i$ at the vertices (x_i, y_i, z_i) of the tetrahedron (labeled $i = 1, 2, 3$), we obtain a system of equations for the unknown coefficients A , B , and C :

$$\begin{aligned} A(x_1 - x_0) + B(y_1 - y_0) + C(z_1 - z_0) &= U_1 - U_0 \\ A(x_2 - x_0) + B(y_2 - y_0) + C(z_2 - z_0) &= U_2 - U_0 \\ A(x_3 - x_0) + B(y_3 - y_0) + C(z_3 - z_0) &= U_3 - U_0 \end{aligned}$$

The determinant of this system is

$$\begin{vmatrix} x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ x_2 - x_0 & y_2 - y_0 & z_2 - z_0 \\ x_3 - x_0 & y_3 - y_0 & z_3 - z_0 \end{vmatrix} = 2V_{tetr},$$

where $V_{tetr} > 0$ is the tetrahedron volume; i.e., the coefficients of the system are uniquely determined. Both derivatives inside the tetrahedron and the flux components (4) are constant.

$$W^x = \frac{\partial U}{\partial x} = A, \quad W^y = \frac{\partial U}{\partial y} = B, \quad W^z = \frac{\partial U}{\partial z} = C. \quad (5)$$

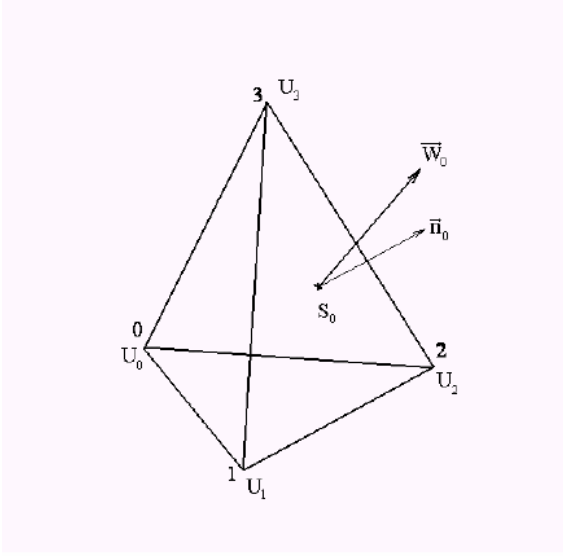


Figure 1: Grid primitive (tetrahedron) and used designation.

Integrating equation (3) over the barycentric control volume Ω centered at the vertex (x_0, y_0, z_0) , we obtain

$$\iiint_{\Omega} \left(\frac{\partial}{\partial x} W^x + \frac{\partial}{\partial y} W^y + \frac{\partial}{\partial z} W^z \right) dx dy dz = 0.$$

Applying the divergence theorem, we write

$$\iiint_{\Omega} \left(\frac{\partial}{\partial x} W^x + \frac{\partial}{\partial y} W^y + \frac{\partial}{\partial z} W^z \right) dx dy dz = \iint_{\partial\Omega} (\vec{W}, \vec{n}) dS. \quad (6)$$

Here, \vec{n} is the outward normal to $\partial\Omega$ and dS is an element of surface area. In the tetrahedron depicted in Fig. 1, S_0 is the area of the face opposite to the vertex labeled 0 (S_i is the area of the face opposite to the i th vertex).

The components of the flux $\vec{W} = (A, B, C)$ can be calculated in terms of normal vectors to the tetrahedron's faces and the corresponding face areas:

$$W^x = \frac{\partial U}{\partial x} = -\frac{1}{3V_{tetr}} \sum_{i=0}^3 S_i \frac{(\vec{n}_i, \vec{e}_x)}{|\vec{n}_i|} U_i,$$

$$W^y = \frac{\partial U}{\partial y} = -\frac{1}{3V_{tet}} \sum_{i=0}^3 S_i \frac{(\vec{n}_i, \vec{e}_y)}{|\vec{n}_i|} U_i, \quad (7)$$

$$W^z = \frac{\partial U}{\partial z} = -\frac{1}{3V_{tet}} \sum_{i=0}^3 S_i \frac{(\vec{n}_i, \vec{e}_z)}{|\vec{n}_i|} U_i,$$

where $\vec{e}_x, \vec{e}_y, \vec{e}_z$ are the global coordinate system's unit basis vectors and $|\vec{n}_i| = 2S_i$. Substituting these expressions for flux components into (6), we obtain the following finite volume approximation:

$$\frac{1}{3} \sum_{ST_0} \left(W_0^x \frac{S_0(\vec{n}_0, \vec{e}_x)}{|\vec{n}_0|} + W_0^y \frac{S_0(\vec{n}_0, \vec{e}_y)}{|\vec{n}_0|} + W_0^z \frac{S_0(\vec{n}_0, \vec{e}_z)}{|\vec{n}_0|} \right)_{ST_0} = 0, \quad (8)$$

where ST_0 is the set of all tetrahedra adjacent to the vertex labeled 0, the summands contain the flux components expressed in terms of the values of U at the vertices of the tetrahedra as (7), S_0 is the area of the face opposite to the vertex labeled 0 in each tetrahedron in ST_0 , and \vec{n}_0 is the corresponding normal.

In the Dirichlet problem, the values of U_i at the vertices lying on the domain boundary are prescribed. In the third boundary value problem, equation (8) written for a boundary vertex contains the fluxes across the faces of the tetrahedra in S_γ belonging to the boundary. In the discretized equation, these fluxes are replaced by the fluxes determined from boundary conditions of the form

$$\frac{\partial U}{\partial \vec{n}} + \sigma(U - \bar{U}) = 0,$$

where σ is a coefficient and \bar{U} is a function prescribed on the boundary. The resulting equation contains additional terms of the form

$$\frac{1}{3} \sigma (U_0 - \bar{U}_0) \sum_{\partial\Omega \in \gamma} S_\gamma.$$

Collecting the coefficients of U_i at the vertices of the tetrahedra in ST_0 , we obtain a system of algebraic equations.

The constructed definite operator corresponds to a finite element approximation operator and so this operator is self-adjoint and positive definite.

When both χ and φ are piecewise continuous functions, a front-tracking tetrahedral grid can be generated and the values of χ inside each tetrahedron T_k are set equal to $\chi = \chi_k$ at the barycenter of the tetrahedron. When equation (2) is inhomogeneous, the right-hand sides of the algebraic equations in (1) are calculated as

$$F_0 = \frac{1}{4} \sum_{T_k \in ST_0} f_k V_k,$$

where f_k is calculated at the barycenter of T_k .

3. VMICCG AND VMICCG WITH REGULARIZED PRECONDITIONER

Equation (1) can be rewritten as

$$a_{ii}y_i - \sum_{j \neq i} a_{ij}y_j = f_i, \quad i = 1, 2, \dots, N, \quad (9)$$

where a_{ii} and $-a_{ij}$ with $j \neq i$ are the elements of the matrix A . When seeking the solution to a 2D boundary value problem for equation (2), we assume that

$$a_{ii} - \sum_{j \neq i} a_{ij} \geq 0, \quad i = 1, 2, \dots, N, \quad (10)$$

$$a_{ij} \geq 0 \text{ for } i = 1, 2, \dots, N, j = 1, 2, \dots, N, j \neq i. \quad (11)$$

Both VMICCG and VMICCG with regularized preconditioner use preconditioned conjugate gradient algorithms of the form⁹

$$r^0 = Ay^0 - f;$$

for $k = 0, 1, \dots$

$$Bw^k = r^k,$$

$$\beta_k = \frac{(w^k, r^k)}{(w^{k-1}, r^{k-1})} \text{ if } k \neq 0, \quad \beta_0 = 0, \quad p^k = w^k + \beta_k p^{k-1}, \quad (12)$$

$$\alpha_k = \frac{(w^k, r^k)}{(p^k, Ap^k)}, \quad y^{k+1} = y^k - \alpha_k p^k, \quad r^{k+1} = r^k - \alpha_k Ap^k,$$

where y^k is the approximate solution of equation (1) obtained at the k th iteration step, B is the preconditioner matrix, and $(u, v) = \sum_{i=1}^N u_i v_i$ is the dot product of the vectors u and v .

In VMICCG, the preconditioner is

$$B = (D^{-1} + A^-)D(D^{-1} + (A^-)^T),$$

where A^- is the strictly lower triangular part of A . The elements d_i of the diagonal matrix D are determined by the condition $Ae + \Lambda D_A e = Be$, where Λ is a diagonal matrix whose elements $\sigma_i \geq 0$ are small and D_A is the diagonal part of A , $e = (1, 1, \dots, 1)^T$, and are calculated as

$$d_i^{-1} = a_{ii}(1 + \sigma_i) - \sum_{k < i} a_{ik} d_k^1, \text{ where } d_k^1 = d_k \sum_{j > k} a_{kj}. \quad (13)$$

To solve system (1) for a 2D problem by the single-processor version of VMICCG, we use the Cuthill–McKee (CM) and reverse Cuthill–McKee (RCM) orderings¹⁰ to improve the convergence rate of the method.

The parameters σ_i in (13) are calculated as

$$\sigma_i = \begin{cases} \frac{\max(1-\rho_i, 0)}{(\rho_i+1)} \alpha h + \alpha^2 h^2 / 2 & \text{if } \sum_{k > i} a_{ik} \neq 0, \\ \alpha^2 h^2 / 2 & \text{if } \sum_{k > i} a_{ik} = 0, \end{cases} \quad (14)$$

or

$$\sigma_i = \begin{cases} \frac{|1-\rho_i|}{(\rho_i+1)} \alpha h + \alpha^2 h^2 / 2 & \text{if } \sum_{k > i} a_{ik} \neq 0, \\ \alpha^2 h^2 / 2 & \text{if } \sum_{k > i} a_{ik} = 0. \end{cases} \quad (15)$$

Here,

$$\rho_i = \sum_{k < i} a_{ik} / \sum_{k > i} a_{ik} \text{ if } \sum_{k > i} a_{ik} \neq 0, \quad (16)$$

$h = \sqrt{S_6}$, S_6 is the Dirichlet cell area in the case of a uniform triangular grid, and $h = \sqrt{S/N}$ in the case of an unstructured triangular grid (S is the area of

the computational domain). The parameter $\alpha > 0$ is determined by optimizing the estimated number of iterations required to compute the Dirichlet problem for Poisson's equation on the original computational domain discretized into a uniform triangular grid or domain close to it¹¹:

$$\alpha = \sqrt{\lambda_1/\sqrt{3}}, \text{ with } 0 < \lambda_1 \sum_i y_i^2 S_6 \leq (A_1 y, y), \quad (17)$$

where A_1 is the coefficient matrix of the system approximating the Dirichlet problem for Poisson's equation. Note that λ_1 in (17) can be evaluated asymptotically by the steepest descent method¹². Our calculations show that λ_1 and α are very slowly varying functions of N . Since the number of iterations exhibits a consistent dependence on α supported by computations, λ_1 and α can be evaluated by using a coarse grid.

When equation (2) is discretized on a tetrahedral grid, the coefficients a_{ij} of system (9) satisfy conditions (10), but conditions (11) are generally violated. For this reason, system (1) is solved by VMICCG with regularized preconditioner. We represent the matrix A as $A = \bar{A} + \tilde{A}$, where the elements \bar{a}_{ii} and $-\bar{a}_{ij}$ ($j \neq i$) of the matrix \bar{A} satisfy the conditions

$$\bar{a}_{ii} - \sum_{j \neq i} \bar{a}_{ij} \geq 0 \quad \text{for } i = 1, 2, \dots, N, \quad (18)$$

$$\bar{a}_{ij} \geq 0 \text{ for } i = 1, 2, \dots, N, j = 1, 2, \dots, N, j \neq i, \quad (19)$$

and the off-diagonal elements $-\tilde{a}_{ij}$ of the matrix \tilde{A} equal the positive off-diagonal elements of A , $\tilde{a}_{ii} = -\sum_{j \neq i} \tilde{a}_{ij}$. Note that $\bar{A} = \bar{A}^T > 0$ and $\tilde{A} = \tilde{A}^T \leq 0$. In VMICCG with regularized preconditioner,

$$B = (D^{-1} + \bar{A}^-)D(D^{-1} + (\bar{A}^-)^T),$$

where \bar{A}^- strictly lower triangular part of \bar{A} . The elements of the diagonal matrix D are determined by the condition $\bar{A}e + \Lambda D_{\bar{A}}^{-1}e = Be$, where Λ is a diagonal matrix whose elements $\sigma_i \geq 0$ are small and $D_{\bar{A}}^{-1}$ is the diagonal part of \bar{A} , and are calculated as

$$d_i^{-1} = \bar{a}_{ii}(1 + \sigma_i) - \sum_{k < i} \bar{a}_{ik}d_k^1, \text{ where } d_k^1 = d_k \sum_{j > k} \bar{a}_{kj}. \quad (20)$$

We calculate σ_i as

$$\sigma_i = \begin{cases} \frac{\max(1-\rho_i, 0)}{(\rho_i+1)} \alpha h + \alpha^2 h^2 / 2 & \text{if } \sum_{k > i} \bar{a}_{ik} \neq 0, \\ \alpha^2 h^2 / 2 & \text{if } \sum_{k > i} \bar{a}_{ik} = 0, \end{cases} \quad (21)$$

or

$$\sigma_i = \begin{cases} \frac{|1-\rho_i|}{(\rho_i+1)} \alpha h + \alpha^2 h^2 / 2 & \text{if } \sum_{k > i} \bar{a}_{ik} \neq 0, \\ \alpha^2 h^2 / 2 & \text{if } \sum_{k > i} \bar{a}_{ik} = 0, \end{cases} \quad (22)$$

where

$$\rho_i = \sum_{k < i} \bar{a}_{ik} / \sum_{k > i} \bar{a}_{ik} \text{ if } \sum_{k > i} \bar{a}_{ik} \neq 0, \quad (23)$$

$h = \sqrt[3]{V_h}$, V_h is the average Dirichlet cell volume in a nearly uniform tetrahedral grid and $h = \sqrt[3]{V/N}$ in the case of an unstructured triangular grid (V is the computational domain volume). The parameter $\alpha > 0$ is evaluated in Section 5.

Vertex ordering in a tetrahedral grid suitable for single-processor computations is not addressed in this study. The numerical solution of a model problem presented in the last section was obtained by successfully using the RCM ordering.

To proceed with theoretical analysis, we need the coefficient matrix \check{A} , with elements \check{a}_{ii} and $-\check{a}_{ij}$ ($j \neq i$), of the system of equations obtained as an approximation of equation (2) before eliminating the unknowns on the computational domain boundary.

L e m m a 1. *Suppose that the matrix \check{A} is such that $\check{a}_{ii} = \sum_{j \neq i} \check{a}_{ij}$ ($i = 1, 2, \dots, N$) and, for any grid point labeled k th such that the corresponding stencil has points lying on the computational domain boundary γ , it holds that $\sum_{j \in \Omega_1^k} |\check{a}_{kj}| \leq \theta \sum_{j \in \Omega_2^k} \check{a}_{kj}$ where $0 < \theta < 1$ is independent of N , $\Omega_1^k = \{j \in \gamma : \check{a}_{kj} < 0\}$, $\Omega_2^k = \{j \in \gamma : \check{a}_{kj} > 0\}$. The elements of A satisfy the following conditions: each row i contains $0 \leq p_i \leq \hat{p}$ positive off-diagonal elements $-a_{i,s_j}$ ($j = 1, \dots, p_i$), and there exist m_{i,s_j} pairs of $-a_{i,k_n}$ and $-a_{k_n,s_j}$ ($n = 1, \dots, m_{i,s_j}$) such that*

$$(a_{i,k_n} \alpha > |a_{i,s_j}|) \wedge (a_{s_j,k_n} \alpha > |a_{i,s_j}|) \quad (24)$$

for each $a_{i,s_j} < 0$, where $0 < \alpha < m/(4\hat{p})$ and α is independent of N . Then,

$$\beta \bar{A} \leq A \leq \bar{A}, \quad (25)$$

where $\beta = \min(1 - 4\hat{p}\alpha/m, 1 - \theta)$ is independent of N .

Note that the last condition means that if an edge (i, s_j) in the graph of A corresponds to $a_{i,s_j} < 0$, then there exist m_{i,s_j} pairs of edges (i, k_n) and (k_n, s_j) ($n = 1, \dots, m_{i,s_j}$) such that conditions (24) are satisfied. This is the case when each element of A contains relatively few positive elements and their absolute values are not too large.

P r o o f is given here in abridged form. We make use of the formula⁵

$$(Ay, y) = \sum_{i=1}^N \sum_{k=i+1}^N a_{ik}(y_i - y_k)^2 + \sum_{k=1}^N (a_{kk} - \sum_{j \neq k} a_{kj}) y_k^2,$$

valid for any symmetric matrix A , where $y \neq 0$ is an arbitrary vector. The assumptions of the lemma are used to prove that

$$(1 - \frac{2\hat{p}2\alpha}{m}) \sum_{i=1}^N \sum_{k=i+1}^N \bar{a}_{ik}(y_i - y_k)^2 \leq \sum_{i=1}^N \sum_{k=i+1}^N a_{ik}(y_i - y_k)^2 \leq \sum_{i=1}^N \sum_{k=i+1}^N \bar{a}_{ik}(y_i - y_k)^2, \quad (26)$$

$$(1 - \theta) \sum_{k=1}^N (\bar{a}_{kk} - \sum_{j \neq k} \bar{a}_{kj}) y_k^2 \leq \sum_{k=1}^N (a_{kk} - \sum_{j \neq k} a_{kj}) y_k^2 \leq \sum_{k=1}^N (\bar{a}_{kk} - \sum_{j \neq k} \bar{a}_{kj}) y_k^2.$$

The left-hand part of inequality (26) is proved by using the representation

$$\begin{aligned} \sum_{i=1}^N \sum_{k=i+1}^N a_{ik}(y_i - y_k)^2 &= 0.5[\sum_{i=1}^N \sum_{k \notin \cup_j K(i,s_j) \cup Z(i), k \neq i} a_{ik}(y_i - y_k)^2 + \\ &\sum_{i=1}^N \sum_{k \in \cup_j K(i,s_j)} a_{ik}(y_i - y_k)^2 + \sum_{i=1}^N \sum_{k \in Z(i)} a_{ik}(y_i - y_k)^2], \end{aligned}$$

where $K(i, s_j) = \{k \neq i, s_j : (a_{i,k} \alpha > |a_{i,s_j}|) \wedge (a_{s_j,k} \alpha > |a_{i,s_j}|)\}$, $Z(i) = \{k \neq i : a_{ik} < 0\}$. The sums are rearranged, and use is made of the fact that each a_{ik} in the second sum corresponds to no more than p_i coefficients a_{ik} in the third sum and no more than p_k coefficients $a_{ik} = a_{ki}$ in the third sum. This completes the proof.

Note that the assumptions of Lemma 1 hold in the 3D problem 5 considered in last section.

4. PARALLEL VERSIONS OF VMICCG AND VMICCG WITH REGULARIZED PRECONDITIONER

To construct VMICCG and VMICCG with regularized preconditioner, the computational domain is decomposed into subdomains containing approximately equal numbers of grid points and the grid points are reordered. We use grid ordering DDO type. This approach makes it possible to overcome the principal difficulty of methods using factored preconditioners, which lies in the recursive calculations required to invert the preconditioner.

A 2D domain is decomposed either by twice applying a procedure analogous to parallel dissections for irregular grids¹⁰ (Method 1) or by hierarchical graph decomposition algorithm¹³ (Method 2). In Method 1 the domain is first decomposed into p_1 subdomains, and each subdomain is then decomposed into p_2 subdomains¹¹, as schematized in Fig. 1. This decomposition method facilitates theoretical analysis of convergence of parallel methods on uniform triangular grids. In the general case, a hierarchical graph decomposition algorithm should be employed. 3D domains should also be decomposed by using hierarchical graph decomposition algorithm. However, geometric decomposition methods are also applicable.

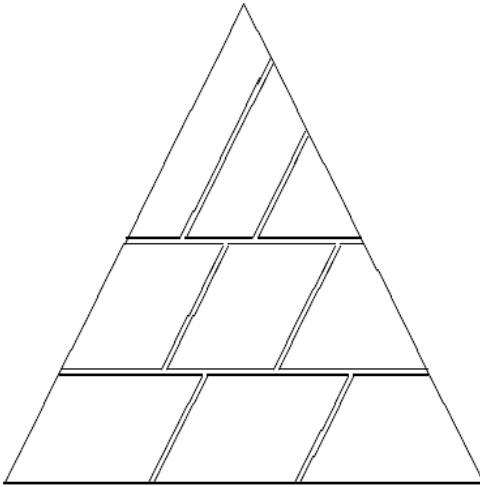


Figure 2: Decomposition of a triangular computational domain into subdomains by Method 1 for $p_1 = p_2 = \sqrt{p} = 3$.

We use an arbitrary subdomain numbering and define a separator set as follows. For the k th subdomain, the grid points on its boundaries with subdomains having higher numbers are considered as separator points; i.e., a grid point on the subdomain boundary belongs to the separator set if at least one point in the corresponding stencil belongs to a subdomain with number $k_p > k$.

To reorder the grid points, the grid points (including the separator points) are first ordered within each subdomain. In 2D computations, the CM or RCM ordering is employed. For 3D problems, either RCM or some other ordering can also be used.

When system (1) is solved by VMICCG with regularized preconditioner, the RCM ordering can be obtained by using the matrix \bar{A} .

We use the following ordering of grid points¹¹. First, the points of grid that do not belong to separators are placed in the order that was established within each subdomain. Then, the separator points are placed, starting from the subdomain with the largest number and following their initial ordering within each subdomain and the descending order of subdomain numbers.

After the rows and columns of A are permuted according to the grid reordering, system (1) is replaced by the system of equations $\hat{A}\hat{y} = \hat{f}$, where $\hat{A} = PAP^{-1}$, $\hat{y} = Py$, $\hat{f} = Pf$, and P is the permutation matrix⁹. In the parallel versions of VMICCG, the preconditioners are defined as follows:

$$\hat{B} = (\hat{D}^{-1} + \hat{A}^-)\hat{D}(\hat{D}^{-1} + (\hat{A}^-)^T), \quad (27)$$

where \hat{A}^- is the strictly lower triangular part of \hat{A} . The elements d_i of the diagonal matrix \hat{D} are determined by the condition $\hat{A}e + GD_{\hat{A}}e = \hat{B}e$, where G is the diagonal matrix whose elements are σ_i , and are calculated by (13) with a_{ii} and $-a_{ij}$ ($i \neq j$) denoting the elements of \hat{A} . The parameters σ_i are calculated by using (14), (16) or (15), (16), where $-a_{ij}$ are the elements of \hat{A} and α is given by (17). In the parallel versions of VMICCG with regularized preconditioner, the preconditioners have the form

$$\hat{B} = (\hat{D}^{-1} + \hat{A}^-)\hat{D}(\hat{D}^{-1} + (\hat{A}^-)^T), \quad (28)$$

where \hat{A}^- is the strictly lower triangular part of \hat{A} , $\hat{A} = P\bar{A}P^{-1}$. The elements d_i of the diagonal matrix \hat{D} are determined by the condition $\hat{A}e + GD_{\hat{A}}e = \hat{B}e$, where G is the diagonal matrix whose elements are σ_i , and are calculated by (20) with $-\bar{a}_{ii}$ and $-\bar{a}_{ij}$ ($i \neq j$) denoting the elements of \hat{A} . The parameters σ_i are calculated by using (21), (23) or (22), (23), where $-\bar{a}_{ij}$ are the elements of \hat{A} . The parameter α is determined by formulas similar to those used in single-processor version of VMICCG with regularized preconditioner (see Section 5).

Parallel computations of equation (1) using VMICCG and VMICCG with regularized preconditioner are performed as follows¹¹. First, the values of d_i and d_i^l are calculated at all grid points by following their ordering within each subdomain, with the exception of the separator points. This procedure does not require any use of data stored in other processors' memories. The values of d_i and d_i^l at separator points are calculated by the following inner iteration process. First, the values of d_i^l calculated at the boundary grid points are sent to the processors that handle the neighboring subdomains with numbers $k_p < k$. Next, d_i (and d_i^l) are calculated at the separator points by following their order within each subdomain if the following conditions hold simultaneously: (1) d_i has not yet been calculated; (2) the stencil centered at the i th grid point does not contain separator points with less numbers j , as ordered within the current subdomain, where d_j has not yet been calculated; (3) the stencil centered at the i th grid point does not contain grid points numbered l from adjacent subdomains with $k_p > k$ where the values of d_l have not been calculated in previous iterations (i.e., d_i^l have not been sent). The iterative process is terminated after all d_i are calculated for all subdomains. It can be shown that the iterative process is convergent. In most 2D problems, no more than three iterations are required if N/p is sufficiently large.

The iterative process of preconditioned conjugate gradient method (12) starts with inversion of the preconditioner, which consists of two steps. For the parallel

version of VMICCG, these steps are as follows: $(\hat{D}^{-1} + \hat{A}^-)\bar{w}^k = (\hat{A}\hat{y}^k - \hat{f})$, $\hat{D}(\hat{D}^{-1} + (\hat{A}^-)^T)w^k = \bar{w}^k$. Analogous steps are performed to invert the preconditioner \hat{B} defined by (28). At the first step of the inversion of \hat{B} , the order of calculation of \bar{w}_i^k follows that of the diagonal elements d_i of \hat{D} . The calculation of w_i^k at the second step is performed in reverse order. The values of w^k at the boundary points of all neighboring subdomains must be received from the respective neighboring processors. Parallelization of the remaining steps of the conjugate gradient method presents no difficulty.

Note that the amount of data transferred per iteration of the conjugate gradient method is larger by a factor of approximately 1.5 in the proposed parallel methods as compared to the parallelized nonpreconditioned conjugate gradient method. The number of times that data transfer is initialized increases when the inner iteration process is executed.

5. THEORETICAL ANALYSIS OF CONVERGENCE RATE OF PARALLEL VERSIONS OF VMICCG AND VMICCG WITH REGULARIZED PRECONDITIONER

It is well known that the number of iterations of preconditioned conjugate gradient method (12) can be estimated as follows⁵:

$$n \geq n_0 = 0.5 \ln(2/\varepsilon) \sqrt{\lambda_{\max}(B^{-1}A)/\lambda_{\min}(B^{-1}A)}, \quad (29)$$

where ε is the desired relative error, and $\lambda_{\max}(B^{-1}A), \lambda_{\min}(B^{-1}A)$ are the highest and lowest eigenvalues of the matrix $B^{-1}A$. Thus, to find estimates for the convergence rates of the methods considered here, the condition number $H(\hat{B}^{-1}\hat{A}) = \lambda_{\max}(\hat{B}^{-1}\hat{A})/\lambda_{\min}(\hat{B}^{-1}\hat{A})$ of the matrix $\hat{B}^{-1}\hat{A}$ must be estimated.

First, we estimate the number of iterations of the parallel version of VMICCG for 2D model problems. As model problem 1, we consider the Dirichlet problem for Poisson's equation on a regular triangle partitioned into a uniform triangular grid. The domain decomposition is performed by Method 1. CM or RCM ordering of grid points is used within each subdomain. A theoretical analysis of the convergence rate of the parallel version of VMICCG with parameters given by (15) and (16) as applied to model problem 1 was presented in¹¹. The convergence rate of the parallel version of VMICCG with parameters given by (14) and (16) is analyzed analogously.

T h e o r e m 1. *The number of iterations of the parallel version of VMICCG (12), (27), (13) with parameters given by (14), (16) or (15), (16) required to compute model problem 1 satisfies the estimate*

$$n \geq n_0 = 0.5 \ln \frac{2}{\varepsilon} \sqrt{(2 + \alpha\psi(p_1, p_2)) \left(1 + \frac{1}{\alpha\sqrt{S_6}}\right)} \approx 0.5 \ln \frac{2}{\varepsilon} \sqrt{\frac{2 + \alpha\psi(p_1, p_2)}{\alpha\sqrt{S}} \sqrt[4]{N}}, \quad (30)$$

where $\psi(p_1, p_2) = c \max(p_1, p_2) \geq 0$ is independent of N , $c = \text{const} > 0$, and α and $\lambda_1 > 0$ are determined by (17).

For model problem 1, a theoretical analysis of the convergence rate of the single-processor version of VMICCG with parameters σ_i given by (14), (16) or (15), (16) is performed in a similar manner. Estimate (30) holds, where ψ is a bounded function independent of N .

Consider the Dirichlet problem with zero boundary conditions for Poisson's equation on an arbitrary convex simply connected domain partitioned into a uniform triangular grid. The domain decomposition is performed by any method (say, 1 or 2) so that all subdomains are convex. We refer to this problem as model problem 2. CM or RCM ordering of grid points is used within each subdomain. A theoretical analysis of the convergence rate of the parallel version of VMICCG with parameters given by (15) and (16) as applied to model problem 2 was presented in¹¹. The convergence rate of the parallel version of VMICCG with parameters given by (14) and (16) is analyzed analogously.

Theorem 2. *The number of iterations of the parallel version of VMICCG (12), (27), (13), with parameters given by (14), (16) or (15), (16) required to compute model problem 2 satisfies the estimate*

$$n \geq n_0 = 0.5 \ln(2/\varepsilon) \sqrt{(2 + \alpha \tilde{\psi}(p)) \left(1 + \frac{1}{\alpha \sqrt{S_6}}\right)} \approx 0.5 \ln(2/\varepsilon) \sqrt{\frac{2 + \alpha \tilde{\psi}(p)}{\alpha \sqrt{S}}} \sqrt[4]{N}, \quad (31)$$

where $\tilde{\psi}(p) \geq 0$ is a monotonically increasing function independent of N and bounded for constant p , and α is given by (17).

For model problem 2, a theoretical analysis of the convergence rate of the single-processor version of VMICCG with parameters σ_i given by (14), (16) or (15), (16) is performed in a similar manner. Estimate (31) holds, where $\tilde{\psi}$ is a bounded function independent of p and N .

According to estimates (30) and (31), the number of iterations required for convergence of VMICCG and parallel version of VMICCG as applied to model problems 1 and 2 is $O(\ln(2/\varepsilon) \sqrt[4]{N})$ for fixed p or p_1, p_2 . Note that estimate (31) is obtained for an arbitrary domain decomposition. When decomposition is performed by Method 1 or 2, it yields an overestimate.

Consider the Dirichlet problem for equation (2) with variable coefficients $\chi_1 = \chi_2 = \chi$ such that $c_0 \leq \chi \leq \bar{c}_0$. Let χ be a piecewise continuous function whose points of discontinuity lie on a finite number of smooth curves. Suppose that both computational domain and all of its subdomains obtained by decomposition are convex and simply connected and CM or RCM ordering of grid points is used within each subdomain and in the entire domain. Then, it can be shown that the number of iterations required for convergence of VMICCG and parallel version of VMICCG with parameters (14), (16) or (15), (16) is $O(\ln(2/\varepsilon) \sqrt[4]{N})$ for fixed p .

Now, we analyze the rate of convergence of the parallel version of VMICCG with regularized preconditioner. The 3D model problem 3, we consider the Dirichlet problem with zero boundary conditions for Poisson's equation on a pyramid. Equation (2) is discretized on a nearly uniform tetrahedral grid by the method described in². We assume that all grid points belong to a set of parallel planes separated by distances of order $O(h)$. Suppose that the points of the stencil centered at any grid point i lie only in the plane containing the i th grid point and the two neighboring planes from the set of parallel planes. For every grid point i , with the possible exception of a number of grid points lying in a finite number of planes from the set of parallel planes, it holds that $|\sum_{j \in \omega_i^+} \bar{a}_{ij} - \sum_{j \in \omega_i^-} \bar{a}_{ij}| \leq \kappa_0 h^2$ where ω_i^+ and ω_i^- are the sets of points of the stencil centered at the i th grid point that lie in two neighboring parallel planes and $\kappa_0 = const$ irrespective of N . Note that the number of such sets of parallel planes differing by the directions of their respective normal vectors $\vec{n}_1, \vec{n}_2, \dots$ ($|\cos(\vec{n}_i, \vec{n}_j)| \neq 1, j \neq i$) can be 1, 2, 3, or 4. Suppose that the matrices A and \bar{A} of a model problem satisfy conditions (25).

Consider a domain decomposition into p subdomains whose boundaries are parallel planes from the set described above or their segments. To simplify analysis, we assume that all subdomain boundaries belong to the same set of parallel planes.

To estimate $\lambda_{\min}(\hat{B}^{-1}\hat{A})$, we assume that the grid orderings within the subdomains satisfy certain conditions that are sufficient and are related to the method of proof. Suppose that the grid ordering used in the parallel version of VMICCG with regularized preconditioner is such that

$$\sigma_i = O(h^2), \text{ for any } i \notin [\cup_{k=1}^p(\omega_k \cup \omega_k^1)] \cup [\cup_{s=1}^t \omega_s] = \Omega, \quad (32)$$

where ω_k is the set of grid points on the subdomain boundaries; ω_k^1 is the set of grid points lying next to the boundary ones; ω_s is the set of grid points in a plane (tentatively labeled s) that belongs to a particular set of planes; $\omega_s \notin \cup_{k=1}^p(\omega_k \cup \omega_k^1)$; and $t = \text{const}$ is independent of N . It is possible that $\cup_{s=1}^t \omega_s = \emptyset$. Suppose also that

$$\frac{\max(1 - \rho_i, 0)}{1 + \rho_i} \leq \theta_1 h \text{ for any } i \in \Omega, \quad (33)$$

where $\theta_1 = \text{const} \geq 0$ is independent of N .

Note that such an ordering can be constructed for model problem 5 (see last section): all grid points are arranged in the order of increasing coordinate x_3 ; then, the grid points with equal coordinates x_3 are arranged in the order of increasing x_1 ; and finally, the grid points with equal x_3 and x_1 are arranged in the order of increasing x_2 . Under this ordering, $\theta_1 = 0$.

It follows from (25) and the equalities $\hat{A} = PAP^{-1}$ and $\hat{\bar{A}} = P\bar{A}P^{-1}$ that

$$\beta(\hat{\bar{A}}y, y) \leq (\hat{A}y, y) \leq (\hat{\bar{A}}y, y), \quad (34)$$

where $y \neq 0$ is an arbitrary vector. To estimate $\lambda_{\max}(\hat{B}^{-1}\hat{A})$, we need the following lemma.

L e m m a 2. *If \hat{B} is the matrix defined by (28) and the diagonal elements of \hat{D} are calculated by (20), where σ_i is given by (21), (23) or (22), (23), then*

$$(\hat{\bar{A}}y, y) \leq (1 + (\alpha h)^{-1})(\hat{B}y, y), \quad (35)$$

where $y \neq 0$ is an arbitrary vector.

P r o o f is analogous to that of Lemma 2 in¹¹.

Combining inequalities (34) and (35) we obtain $(\hat{A}y, y) \leq (1 + (\alpha h)^{-1})(\hat{B}y, y)$ and, therefore,

$$\lambda_{\max}(\hat{B}^{-1}\hat{A}) \leq 1 + (\alpha h)^{-1}. \quad (36)$$

To estimate $\lambda_{\min}(\hat{B}^{-1}\hat{A})$, we need the following two lemmas.

L e m m a 3. *Let $\tilde{\rho}_i \geq 0$ be a grid function defined on the interior of a tetrahedral grid that discretizes a simply connected domain \bar{G} . Suppose that $\tilde{\rho}_i$ is not identically zero. For any grid function y_i defined on this grid in the domain \bar{G} with boundary γ such that $y_i = 0$ on γ , it holds that*

$$\sum_{i=1}^N \tilde{\rho}_i y_i^2 \leq (\max_i v_i)(\bar{A}y, y), \quad (37)$$

where v_i is the solution to the system of equations $\bar{A}v = \tilde{\rho}$, $\bar{A} = \bar{A}^T > 0$, and the elements of \bar{A} satisfy conditions (18) and (19).

P r o o f is analogous to that of Lemma 13 in¹² (p.239).

L e m m a 4. *Let A be the coefficient matrix of an approximation of model problem 3 and the elements of \bar{A} satisfy the conditions $\bar{a}_{ii} \leq c_1 h$, $\bar{a}_{ij} \geq c_2 h$, where $c_1 = \text{const} > 0$ and $c_2 = \text{const} > 0$ are independent of h and N , and conditions (18) and (19). Suppose that the computational domain is decomposed into p subdomains in one spatial direction, as described above, and conditions (32) and (33) for grid ordering are satisfied. If \hat{B} is the matrix defined by (28) and the diagonal elements of \hat{D} are calculated by (20), where σ_i is given by (21) and (23), then*

$$(\hat{A}y, y) \geq (1 + 0.5\alpha^2 c_1/\lambda_1 + \alpha c_1 \theta_1/\lambda_1 + \alpha \bar{\psi}(p))^{-1} (\hat{B}y, y), \quad (38)$$

where $\lambda_1 > 0$ satisfies the inequality

$$\lambda_1 \sum_i y_i^2 h^3 \leq (\hat{A}y, y), \quad (39)$$

$$\bar{\psi}(p) = c_5 [2(p-1) + t], \quad (40)$$

$c_5 = \text{const} > 0$ is independent of N , and $y \neq 0$ is an arbitrary vector.

P r o o f is given in abridged form, because some of the arguments used are analogous to those developed in¹¹. Simple rearrangements yield

$$(GD_{\bar{A}})y, y) \leq (0.5\alpha^2 + \alpha\theta_1)c_1 h^3 (y, y) + \sum_{k=1}^p [\sum_{i \in \omega_k} \alpha h \bar{a}_{ii} y_i^2 + \sum_{i \in \omega_k^1} \alpha h \bar{a}_{ii} y_i^2] + \sum_{s=1}^t \sum_{i \in \omega_s} \alpha h \bar{a}_{ii} y_i^2,$$

where $y \neq 0$ is an arbitrary vector. The first summand is estimated by using inequality (39). An estimate for $\sum_{i \in \omega_k} \alpha h \bar{a}_{ii} y_i^2$ is based on Lemma 3, the function v_i has a bounded majorant $\tilde{v}_i \leq c_5$, where $c_5 = \text{const} > 0$ is independent of N , and the maximum principle¹⁴ is used. The remaining summands are estimated analogously. The result is $(GD_{\bar{A}})y, y) \leq [0.5\alpha^2 c_1/\lambda_1 + \theta_1 \alpha c_1/\lambda_1 + \alpha \bar{\psi}(p)] (\hat{A}y, y)$, where $\bar{\psi}(p)$ is defined by (40). Since the inequality $(\hat{B}y, y) \leq ((\bar{A} + GD_{\bar{A}})y, y)$, proved in¹¹, also holds for 3D problems, we obtain (38), which completes the proof.

Combining (34) with (38), we obtain the inequality

$$(\hat{A}y, y) \geq \beta (1 + 0.5\alpha^2 c_1/\lambda_1 + \alpha c_1 \theta_1/\lambda_1 + \alpha \bar{\psi}(p))^{-1} (\hat{B}y, y),$$

valid for any $y \neq 0$. Hence,

$$\lambda_{\min}(\hat{B}^{-1}\hat{A}) \geq \beta (1 + 0.5\alpha^2 c_1/\lambda_1 + \alpha \theta_1 c_1/\lambda_1 + \alpha \bar{\psi}(p))^{-1}. \quad (41)$$

Let α be determined by the minimum condition for the coefficient of $1/h$ in the estimate for the condition number $H(\hat{B}^{-1}\hat{A})$:

$$\alpha = \sqrt{2\lambda_1/c_1}, \quad (42)$$

where λ_1 satisfies condition (39) and $c_1 = \text{const} > 0$ is independent of N . Note that (39) is equivalent to the condition

$$\lambda_1 \sum_i y_i^2 h^3 \leq (\bar{A}y, y), \quad (43)$$

which is better suited for evaluating λ_1 . Using (29), (36), and (41), we find that the number of iterations of the parallel version of VMICCG with regularized preconditioner as applied to model problem 3 satisfies the estimate

$$n \geq n_0 = \ln \frac{2}{\varepsilon} \sqrt{\frac{2 + \theta_1/\alpha + \alpha\bar{\psi}(p)}{4\beta}} \left(1 + \frac{1}{\alpha h}\right) \approx \ln \frac{2}{\varepsilon} \sqrt{\frac{2 + \theta_1/\alpha + \alpha\bar{\psi}(p)}{4\beta\alpha\sqrt[3]{V}}} \sqrt[6]{N}. \quad (44)$$

In summary, we have the following theorem.

Theorem 3. *The number of iterations of the parallel version of VMICCG with regularized preconditioner defined by (12), (28), (20), with σ_i given by (21) and (23), as applied to model problem 3 satisfies estimate (44), where the function $\bar{\psi}(p) \geq 0$ defined by (40) is independent of h and N ; α is determined by (42); $\lambda_1 > 0$ satisfies inequality (43); and the constants $\theta_1 \geq 0$, $0 < \beta < 1$, $c_1 > 0$, $c_5 > 0$, and $t \geq 0$ are independent of N .*

Note that if the parameters σ_i in the parallel version of VMICCG with regularized preconditioner defined by (12), (28), (20) are calculated by (22) and (23), then the theoretical analysis of the convergence rate is performed analogously.

Note also that the analysis of the convergence rate of the single-processor version of VMICCG with regularized preconditioner as applied to model problem 3 is performed analogously if the grid ordering is such that $\sigma_i = O(h^2)$ at all grid points except those in a finite number t (independent of N) of parallel planes belonging to a particular set of parallel planes and condition (33) is satisfied. Estimate (44) is valid for $p=1$.

Numerical results presented below show that λ_1 is a slowly varying function of N . Moreover, λ_1 is a monotonically increasing function, and therefore so is α . Using (44), we find that the number of iterations required for convergence of the methods is $O(\ln(2/\varepsilon)\sqrt[6]{N})$ for fixed p .

Note that if the computational domain is decomposed in two or three spatial directions corresponding to planes from distinct sets of parallel planes, then analogous estimates for the number of iterations can be obtained, with $\bar{\psi} = O(\sqrt{p})$ or $\bar{\psi} = O(\sqrt[3]{p})$.

When a boundary value problem for equation (2) with variable coefficients χ_1 and χ_2 or a problem discretized on an unstructured tetrahedral grid is computed by VMICCG with regularized preconditioner or its parallel version, α is calculated as in the Dirichlet problem for Poisson's equation on the same domain discretized into a nearly uniform tetrahedral grid.

Consider the Dirichlet problem for equation (2) with coefficients $\chi_1 = \chi_2 = \chi$, where $c_0 \leq \chi \leq \bar{c}_0$ is a piecewise continuous functions whose points of discontinuity lie in a finite number of planes belonging to the sets of planes from the set of parallel planes described above. The solution is sought on a pyramid discretized into a nearly uniform tetrahedral grid. Suppose that the grid, the domain decomposition, and the ordering of grid points satisfy the conditions formulated for model problem 3. Then, the number of iterations of the VMICCG with regularized preconditioner and its parallel version satisfy estimates analogous to (44); i.e., convergence of these methods requires a number of iterations of order $O(\ln(2/\varepsilon)\sqrt[6]{N})$ for fixed p .

6. NUMERICAL RESULTS

Both VMICCG and VMICCG with regularized preconditioner proposed here

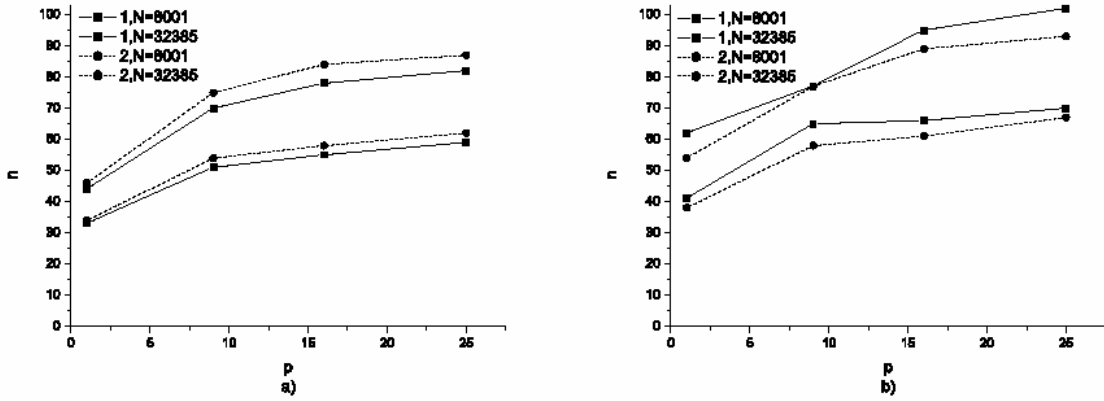


Figure 3: a) Results of calculations of model problem 1. b) Results of calculations of model problem 2.

were used to solve five model problems having exact solutions. To determine the number of necessary iterations, computations were continued until the condition $(Ay^k - f, y^k - y) \leq \varepsilon^2(Ay^0 - f, y^0 - y)$ was fulfilled, where $y^0 \equiv 0$ and $\varepsilon = 10^{-8}$ for problems 1-4 and $\varepsilon = 10^{-6}$ for problem 5. The computations were performed on MVS 1000M and MVS 15000 parallel computers.

We solved Dirichlet problems for equation (2) on a regular triangle with vertex coordinates $(-1,-1)$, $(1,-1)$, and $(0, \sqrt{3} - 1)$. In problem 1, $\chi_1 = \chi_2 = 1$; in problem 2, $\chi_1 = \chi_2 = \chi = 100000$, if $x_2 > -0.5$ and $|x_1| < 0.25$, and $\chi_1 = \chi_2 = \chi = 1$ otherwise.

The elliptic equation was discretized on a uniform triangular grid¹ with $N = 8001, 32385$. The right-hand side f was such that the exact solution to equation (1) was $y = 8.2(x_1 + 1.1)(1.1 - x_1)(x_2 + 1.09)$. The domain decomposition was performed by Method 1 (see Section 4), with $p_1 = p_2 = \sqrt{p}$. $\lambda_1 \approx 26.30$ and $\alpha = 3.90$ were calculated for $N = 8001$.

Figures 3 a) and 3 b) show the number of iterations required to solve problems 1 and 2, respectively, as a function of the number of processors for different values of N . Curves 1 correspond to computations by the parallel version of VMICCG, with σ_i calculated by using (14) and (16). Curves 2 correspond to computations by the parallel version of VMICCG, with σ_i calculated by using (15) and (16). In all computations, the number of iterations was approximately proportional to $\sqrt[4]{N}$ and increased with the number of processors by a factor less than two. The number of iterations required to solve problem 1 was smaller when σ_i was calculated by using (14) and (16). The number of iterations required to solve problem 2 was smaller when σ_i was calculated by using (15), (16).

As Problems 3 and 4, we considered the Dirichlet problem for Poisson's equation on a unit square. Poisson's equation was discretized on unstructured triangular grids progressively refined from periphery to center of the domain, with a refinement ratio of 5 in problem 3 and 100 in problem 4. The grids were generated by the method described in¹⁵. Note that the resulting matrix A had a small number of positive off-diagonal elements. The right-hand side of f was such that the exact solution to equation (1) was $y = 32x_1(1 - x_1)x_2(1 - x_2)$. The computational domain was decomposed into subdomains by using a hierarchical graph decomposition algorithm. λ_1 and α were evaluated for the Dirichlet problem for Poisson's equation on a uniform

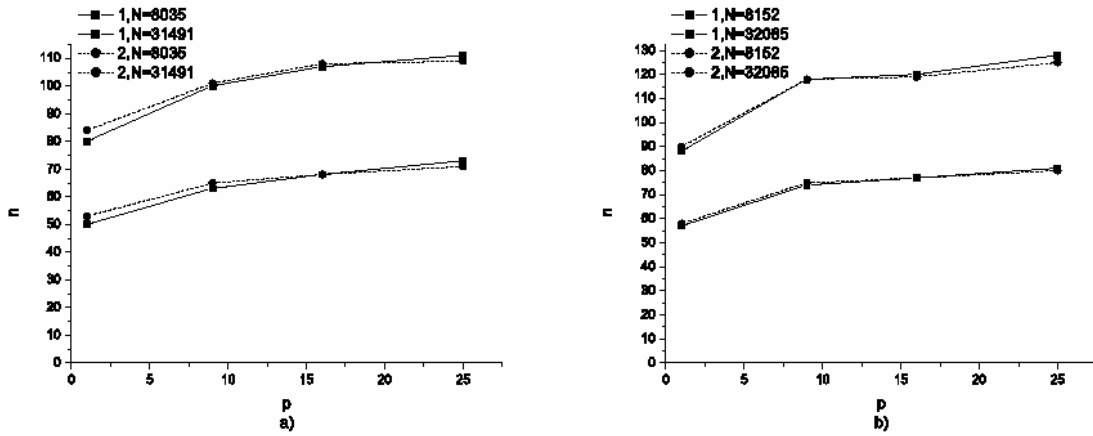


Figure 4: a) Results of calculations of model problem 3. b) Results of calculations of model problem 4.

triangular discretization of a nearly square domain with $N = 1600$: $\lambda_1 \approx 38.7$, $\alpha = 4.73$.

Figures 4 a) and 4 b) show the number of iterations required to solve problems 3 and 4, respectively, as a function of the number of processors for several values of N . Curves 1 correspond to computations by the parallel version of VMICCG, with σ_i calculated by using (14) and (16). Curves 2 correspond to computations by the parallel version of VMICCG, with σ_i calculated by using (15) and (16). In all computations, the number of iterations increased with N faster than $\sqrt[4]{N}$ and was approximately proportional to N^β , where $0.28 < \beta < 0.35$. In all computations, the number of iterations increased with the number of processors by a factor less than two.

The parallel versions of VMICCG with regularized preconditioner proposed in this study were used to solve problem 5, the 3D Dirichlet problem for Poisson's equation on an irregular pyramid with vertices at the points $(0,0,0)$, $(2,1,0)$, $(2,-2,0)$, $(1,0,4)$. The elliptic equation was discretized on nearly uniform tetrahedral grids generated by the method described in², with $N = 455, 4495, 39711, 333375$. The right-hand side f was such that the exact solution to equation (1) was $y = x_3(8 - 4x_1 - x_3)(8x_2 - 4x_1 + x_3)(x_3 - 4x_1 - 4x_2)$, which vanishes on the domain boundary. The domain was decomposed into subdomains by a geometric method. For $p = 4$, the pyramid was cut by the plane $x_3 = 0.8252$ parallel to its base and the plane defined by the points $(0,0,0)$, $(2,-0.5,0)$, $(1,0,4)$, which contains an edge of the pyramid and passes through the midpoint of the nonadjacent edge. For $p = 8$, the domain was cut by the plane $x_3 = 0.8252$; the plane defined by the points $(0,0,0)$, $(2,-0.5,0)$, $(1,0,4)$; and the plane defined by the points $(1.4142, 0.7071, 0)$, $(1.4142, -1.4142, 0)$, $(1,0,4)$. For $p = 18$, the domain was cut by the planes $x_3 = 1.227$ and $x_3 = 0.5057$; the planes passing through the points $(0,0,0)$, $(2,0,0)$, $(1,0,4)$ and $(0,0,0)$, $(2,-1,0)$, $(1,0,4)$; and the plane defined by the points $(1.4142, 0.7071, 0)$, $(1.4142, -1.4142, 0)$, $(1,0,4)$. RCM ordering was used in the entire computational domain and within each subdomain. Using the steepest descent method, we found that $\lambda_1 \approx 17.68$ for $N = 455$, $\lambda_1 \approx 18.58$ for $N = 4495$, $\lambda_1 \approx 19.41$ for $N = 39711$, and $\lambda_1 \approx 22.28$ for $N = 333375$. For problem 5, $c_1 = 1.13$.

Figure 5 shows the number of iterations required to solve problem 5 as a function

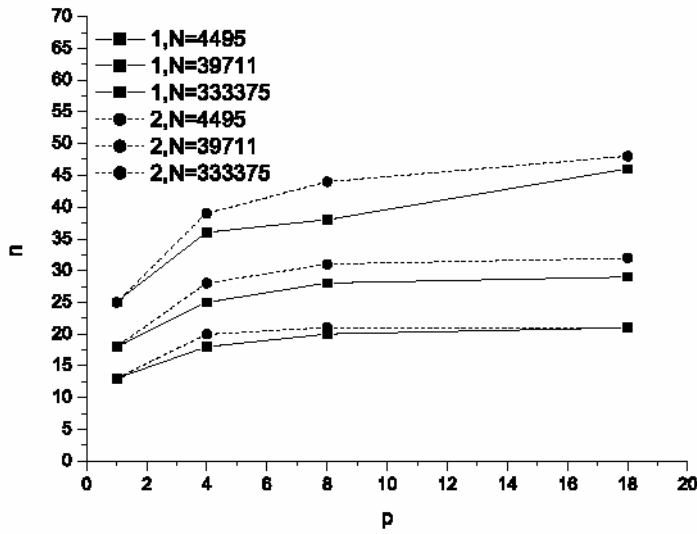


Figure 5: Results of calculations of model problem 5.

of the number of processors for several values of N . Curves 1 correspond to computations by the parallel version of VMICCG with regularized preconditioner, with σ_i calculated by using (21) and (23). Curves 2 correspond to computations with σ_i calculated by using (22) and (23). In all computations, the number of iterations was approximately proportional to $\sqrt[6]{N}$ and increased with the number of processors by a factor less than two. Note that the number of necessary iterations was smaller when σ_i was calculated by using (21), (23) as compared to (22), (23). If $\sigma_i = 0.5\alpha^2 h^2$ for all i then the number of iterations does not exhibit asymptotic behavior as a function of the number of grid points in the parallel version of VMICCG with regularized preconditioner. It should be noted that solution of problem 5 by the single-processor version of VMICCG without regularized preconditioner required a similar number of iterations. However, an attempt of parallelization with parameters σ_i calculated by using (14), (16) and (15), (16) even on 8 processors resulted in a rapid increase in the number of iterations with N .

In summary, we have proposed parallel versions of VMICCG (for 2D problems) and VMICCG with regularized preconditioner (for 3D problems) based on grid ordering of DDO type. Analyzing model problems as examples, we have proved and numerically demonstrated that the number of iterations required for convergence of the proposed parallel methods is $O(\ln(2/\varepsilon)\sqrt[4]{N})$ for fixed p in 2D problems solved on uniform triangular grids and $O(\ln(2/\varepsilon)\sqrt[6]{N})$ in 3D problems solved on nearly uniform tetrahedral grids. The increase in the number of iterations with the number of processors is acceptable for moderate number of processors. This guarantees good efficiency of the proposed methods in the case of a moderate number of messages.

This work was supported by the Russian Foundation for Basic Research, project nos. 05-07-90230 and 06-01-00233.

REFERENCES

- [1] I.V. Popov, S.V. Polyakov, "Construction of adaptive irregular triangular grids

- for two-dimensional multiply connected nonconvex domains", *Mat. Model*, Vol.14, N5, 25-35 (2002).
- [2] I.V. Popov, "A new approach to constructing tetrahedral grids", *Fundamental Problems of Mathematical Physics and Modeling Technical and Technological Systems*, Moscow, 10, 128-133 (2007).
- [3] I.V. Popov, S.V. Polyakov, Yu.N. Karamzin, "High Accuracy Difference Schemes on Unstructured Triangle Grids", *In Numerical Methods and Applications*, (5th int. conf. NMA 2002, Revised Papers, Ed. I. Divov, I. Lirkov, S. Margenov, Z. Zlatev, Springer, Berlin, 2003.)
- [4] J. A. Meijerink, H. A. van der Vorst, "An Iterative Solution Method for Linear Systems, of which the Coefficient Matrix is a Symmetric M-matrix", *Math. Comp*, Vol.31, N 137, 148-162 (1977).
- [5] I. Gustafsson, "A Class of First Order Factorization Methods", *BIT*, 18, 142-156 (1978).
- [6] D. Kershaw, "The Incomplete Choleski-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations", *J. Comp. Phys*, 26, 43-65 (1978).
- [7] I.E. Kaporin, I.N. Konshin, "Parallel Conjugate Gradient Preconditioning via Incomplete Cholesky of Overlapping Submatrices", *Parallel Computational Fluid Dynamics*, (Moscow, Russia, May 13-15, 2003). Edited by B. Chetverushkin, A. Ecer, J. Periaux and N. Satofuka-Assistant Editor: Pat Fox, Elsevier Science B.V., Amsterdam, 81-88 (2004).
- [8] Yu.V. Vasilevskii, "Parallel Computing Technologies for Boundary Value Problems", *Doctoral Disertation*, Moscow (2006).
- [9] J.M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum, New York, (1988).
- [10] A. George, J. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Englewood Cliffs (U.S.A.): Prentice-Hall, (1973).
- [11] O.Yu. Milyukova, "New parallel iteration methods with factored preconditioning matrices for solving elliptic equations on triangular grids", *Mat. Model*, Vol.19, N9, 27-48 (2007).
- [12] A.A. Samarskii, E.S. Nikolaev, *Methods for Solving Grid Equations*, Moscow, Nauka, (1978) [in Russian].
- [13] M.V. Yakobovskii, *Processing of grid data on distributed computing systems*, Questions of Atomic Science and Technology. Ser. Mathematical Modeling of Physical Processes, 2, 40-53 (2004).
- [14] A.A. Samarskii, *Theory of finite Difference schemes*, Moscow, Nauka, (1989) [in Russian].
- [15] A.V. Tkhir, "The advanced front method for constructing nonstructured grids", *Numerical Methods and Applications*, Ed. by Yu.A. Kuznetsov. Inst. Vychisl. Mat., Ross. Akad. Nauk. (1995).