Parallel Iterative Methods with Factorized Preconditioners for Discrete Elliptic Equations

Olga Yu. Milyukova*

Institute for Mathematical Modelling, Russian Academy of Sciences
4a, Miusskaya sq., Moscow, 125047, Russia

Key words: ordering strategies, parallel computations.

The system of differential equations in pressure-velocity variables is often used for mathematical modelling of hydrodynamic problems. The elliptic eqiation for pressure should be solved. Note, that the main computational costs are addressed to numerical solution of this elliptic equation.

The purpose of this study is to solve the symmetric, positive definite system of equations

$$Ay = f\,, \quad A = A^T > 0\,, \tag{1}$$

obtained from difference approximation of the Dirichlet or Neumann problem for the elliptic equation

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

on distributed-memory parallel computers. In Eq. (2), m=2 and m=3 for 2D and 3D problems, respectively; $C \ge \chi_\alpha \ge c > 0$; $\alpha = 1,2,3$; $d(x) \ge 0$; $x = (x_1, x_2)$ or $x = (x_1, x_2, x_3)$. The computational domain is a rectangle or a rectangular parallelepiped. For two-dimensional problems, connected domains of more complex geometry are also considered, such as an equilateral triangle. In the case of a rectangular domain, Eq. (2) is approximated on both uniform and nonuniform grids or on a locally refined grid. In the case of an equilateral triangle, Eq. (2) is approximated on a uniform triangular grid. 3D problems are approximated on a uniform orthogonal grid.

To solve Eq. (1) on a uniform or nonuniform orthogonal grid, parallel versions of incomplete Cholesky factorization (ICCG(0)) [1], modified incomplete Cholesky factoriza- tion (MICCG(0)) [2], and alternating triangular conjugat e gradient (ATMCG) [3] methods are proposed. While the 2D equation is approximated on a locally refined or uniform triangular grid, it is suggested to solve Eq. (1) by using parallel versions of VICCG and VMICCG (variants of ICCG(0), MICCG(0) mentioned by I.E. Kaporin in the supplement to the Russian translation of [1]).

In VICCG and VMICCG, the preconditioner matrix has the form

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

where A^- is the strictly lower triangular part of A, $A^+ = (A^-)^T$. The choice of the diagonal matrix D is based on the same considerations as in ICCG(0) and MICCG(0).

To solve Eq. (1) on a distributed-memory parallel computer, the computational domain is partitioned into subdomains along two spatial directions, which corresponds to computa- tion on a two-dimensional processor array. The major difficulty in parallelizing the methods mentioned above lies in the recursive procedure used to calculate the inverse of the preconditioner matrix and the elements of the diagonal matrix D. To overcome this difficulty, the grid points are reordered and the preconditioner matrix is reconstructed. Various orderings of domain

*E-mail: miliukova@imamod.ru

decomposition type are used in this study. This approach makes it possible for all processors to execute computations simultaneously in the respective subdomains.

In the parallel versions of ICCG(0), MICCG(0) (implemented on a uniform orthogonal grid), VICCG, and VMICCG, the preconditioner matrix has the form

$$B = (D^{-1} + A_1)D(D^{-1} + A_1^T),$$

where A_1 is not a lower triangular matrix under the original ordering of grid points. In the parallel versions of ICCG(0) and VICCG, the diagonal matrix D is such that the diagonal entries of A and B are equal. In the parallel versions of MICCG(0) and VMICCG, the diagonal matrix D is determined by the condition

$$Ae + \lambda D_A e = Be,$$

where $e = (1,1,...,1)^T$, D_A is the diagonal part of A, and λ is the diagonal matrix with small diagonal entries σ_i. When the coefficients χ_α ($\alpha = 1,2$) are continuous functions, one should use $\sigma_i = O(1/\sqrt{N})$ on the portion of the boundaries between the subdomains where the computation starts and $\sigma_i = O(1/N)$ or $\sigma_i = 0$ at the interior points of the subdomains and on the remaining portions of the boundaries between the subdomains (N is the number of grid points in the entire two-dimensional computational domain). When the parallel version of MICCG(0) is used to solve the 3D problem on a uniform orthogonal grid, one should use, respectively, $\sigma_i = O(1/N_h)$ and $\sigma_i = O(1/N_h^2)$, where N_h is the number of grid points in one spatial direction. In the parallel versions of MICCG(0) applied to the approximation of Eq. (2) on a uniform orthogonal grid and in the parallel version of VMICCG applied to the approximation of Eq. (2) on a uniform triangular grid, the parameters σ_i are determined by minimizing the estimated number of iteration steps in computing the Dirichlet problem for Poisson's equation. In the parallel version of VMICCG applied to the Neumann problem with $d(x) \equiv 1$ implemented on locally refined grids, the parameters σ_i employed were determined for the solution of the finite-difference Dirihlet problem for Poisson's equation on the uniform orthogonal grid with mesh size $\sqrt{S}/\sqrt{N}$, where S is the square of the rectangle.

In the parallel version of ATMCG, the preconditioner matrix has the form

$$B = (D + \omega_0(0.5D_A + A_1))D^{-1}(D + \omega_0(0.5D_A + A_1^T)),$$

where the diagonal matrix D and the iteration parameter ω_0 are determined by minimizing the estimated number of iteration steps.

For 2D and 3D Dirichlet problems with $d(x) \equiv 0$ solved on an orthogonal grid, it was proved in [4,5] that when the coefficients χ_α are sufficiently smooth functions, the parallel versions of MICCG(0) and ATMCG are convergent on any particular processor array if the number of iteration steps is at least $O(\sqrt[4]{N}\ln(2/\varepsilon))$ and $O(\sqrt[6]{N}\ln(2/\varepsilon))$, respectively, where ε is the admissible relative error. The required number of iteration steps slowly increases with the number of processors. For the 2D Dirichlet problem for Poisson's equation approximated on a uniform triangular grid, it is proved that the parallel version of VMICCG is convergent on any particular processor array if the number of iteration steps is at least $O(\sqrt[4]{N}\ln(2/\varepsilon))$. The corresponding number of necessary iteration steps slowly increases with the number of processors.

Computations performed for model problems have shown that the asymptotic behavior of the required number of iteration steps as a function of the number of grid points is similar to that characteristic of the corresponding single-processor versions in all proposed parallel versions implemented on all aforementioned types of grids used to approximate Eq. (2) on any particular processor array. The required number of iteration steps slowly increases with the number of processors. Computations performed on a 32-processor Parsytec CC workstation and on an

MVS-1000M parallel computer with a moderate number of processors demonstrated that the proposed methods are characterized by a good efficiency.

Acknowledgements

This work has been supported by Russian Fund for Basic Research (Grant No. 02-07-90168).

Reference

1. J. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*,} New York: Plenum, (1988).
2. Gustafsson, A class of first order factorization methods, *BIT*, **18** (1978), 142-156.
3. A.A. Samarskii and E.S. Nikolaev, *Numerical Methods for Grid Equations*, Basel: Birkhauser, 1989.
4. O.Yu. Milyukova, Parallel approximate factorization method for solving discrete elliptic equations, *Parallel Computing,* **27** (2001), 1365-1379.
5. O.Yu. Milyukova, A Parallel Version of the Generalized Alternating Triangular Method for Elliptic Equations, *Comput. Math. Math. Phys.*, **38** (1998), No. 12, 2002-2012.