Parallel Conjugate Gradient Preconditioning via Incomplete Cholesky of Overlapping Submatrices

Igor Ye. Kaporin* and Igor N. Konshin

Center for Supercomputer and Massively Parallel Applications
Computing Center of Russian Academy of Sciences
40, Vavilova st., Moscow, 119991, Russia

Key words: conjugate gradients, parallel preconditionings, incomplete Cholesky.

Introduction

We consider the Preconditioned Conjugate Gradient (PCG) algorithm for solving linear algebraic system

$Ax = b$,

where A is large sparse unstructured Symmetric Positive Definite (SPD) coefficient matrix of order n. For the construction of preconditioning, we use the Block Incomplete Inverse Cholesky (BIIC) factorization first proposed in [1] and considered in more detail in [3]. For the purpose of better parallelism, we use the additive form of this preconditioner given in [3].

The BIIC preconditioner is formulated in terms of an s x s block form of the matrix A, where the *exact* Cholesky factorization is used to solve subsystems associated with properly augmented diagonal blocks. When aiming at efficient parallel solution of large problems, the block size will be approximately n/s, where s is usually taken to be equal to the number of processors available. For large problems and with a limited number of processors, these blocks may be still too large to be processed with the exact Cholesky. Fortunately, using here an *incomplete* factorization may yield even more efficient preconditioning. We employ the so-called robust Incomplete Cholesky 2nd order factorization (IC2) method developed in [4]. The use of IC2 significantly reduces the preconditioning cost (both evaluation and application) but almost does not impair the quality of the resulting preconditioner.

Such construction results in a purely algebraic procedure which is applicable to the preconditioning of any sparse SPD matrix independent of the nature of the underlying physical problem and the type of discretization method used.

The conjugate gradient iterations

We use the PCG iterations in the standard form:

$r_0 = b - Ax_0$, $p_0 = Cr_0$;

for $i = 0, 1, \ldots$:

$\alpha_i = r_i^T Cr_i / p_i^T Ap_i$, $x_{i+1} = x_i + p_i\alpha_i$, $r_{i+1} = r_i - Ap_i\alpha_i$,

$\beta_i = r_{i+1}^T Cr_{i+1} / r_i^T Cr_i$, $p_{i+1} = Cr_{i+1} + p_i\beta_i$.

Here C is a properly chosen SPD preconditioning matrix, which should approximate, in some sense, the matrix A^{-1}. The choice of the matrix C is subject to the requirement that the vector $w = Cr$ be easily calculated for any r. For instance, one of the best choices is the approximate Cholesky preconditioning, where $C = (U^T U)^{-1}$ and $U^T U \approx A$ with the upper triangular matrix U being much sparser than the exact Cholesky factor of A, cf. [4] and references therein.

*E-mail: kaporin@ccas.ru

In order to get a constructive approach to preconditioning optimization, one can use the so-called K-condition number

$$K = K(CA) = \left(n^{-1} trace(HA)\right)^n / \det(HA)$$

instead of the standard spectral condition number $\lambda_{\max}(CA)/\lambda_{\min}(CA)$. The underlying PCG iteration number bound [2, 3] is

$$i \le i_K(\varepsilon) = \log_2 K + \log_2 1/\varepsilon ,$$

where i is the iteration number related to the convergence criterion $r_i^T A^{-1} r_i \le \varepsilon^2 r_0^T A^{-1} r_0$. The preconditioning described in the next section is nearly optimum with respect to the K-condition number minimization, see [9].

Parallel preconditioning

For parallel computations, the preconditioner C was constructed using the Block Incomplete Inverse Cholesky (BIIC) [3] and the 2nd order Incomplete Cholesky (IC2) [4] techniques. Let A be reordered and split into an s x s block form using, e.g. techniques described in [5]. For the t-th diagonal block having the dimension n_t, the ``basic'' index set

$$\{k_{t-1}+1, \ldots, k_t\},$$

is defined, where $k_{t-1} = n_1 + \ldots + n_{t-1}$, k_0=0, k_s=n, and the ``overlapping'' index set is constructed as

$$\{j_t(1), \ldots, j_t(m_t - n_t)\}, \quad j_t(p) \le k_{t-1} .$$

The BIIC-IC2 preconditioner \$C\$ is:

$$C = C(\tau) = \sum_{t=1}^{s} V_t U_t^{-1}(\tau) \begin{bmatrix} 0 & 0 \\ 0 & I_{n_t} \end{bmatrix} U_t^{-T}(T) V_t^T ,$$

where V_t are rectangular matrices composed of unit n-vectors e_j as follows:

$$V_t = [e_{j_t(1)} | \ldots | e_{j_t(m_t - n_t)} | e_{k_{t-1}+1} | \ldots | e_{k_t}], \quad t = 1, \ldots, s ,$$

and each upper triangular matrix $U_t(\tau)$ is an incomplete Cholesky factor for the t-th $m_t \times m_t$ submatrix $V_t^T A V_t$:

$$V_t^T A V^T = U_t^T(\tau) U_t(\tau) + U_t^T(\tau) R_t(\tau) + R_t^T(\tau) U_t(\tau)$$

Here $R_t(\tau)$ is a strictly upper triangular error matrix.

For each t, the ``overlapping'' index set typically includes indices not greater than $k_t$ and the most ``essentially'' connected to the basic index set, e.g. in the sense of the sparse matrix graph adjacency relations. Here $m_t \ge n_t$ and, obviously, $m_1 = n_1$, i.e. at least the first overlapping set is empty.

Here $0 < \tau \ll 1$ is the drop tolerance parameter which determines the quality of the incomplete factorization. The existence and correctness of such IC2 decompositions is guaranteed for any SPD matrix [4]. The recurrences for the calculation of IC2 factorization can easily be obtained from the above relation, especially for the case in which the sparsity patterns of U and R do not have coinciding nonzero positions and their nonzero elements are subject to the conditions $U_{ij} \ge \tau$ and $|R_{ij}| < \tau$, respectively, $i < j$, see [4].

Parallel implementation and testing results

The above described mathematical technique was implemented in a portable software with the use of the Message Passing Interface (MPI) library for communications between processes. The special structure of the above described BIIC-IC2 preconditioning made it possible to run the PCG iterations very efficiently even on workstation clusters.

The results of numerical experiments were presented in [6-9]. Due to the additive structure of the preconditioner, the parallel overhead of its implementation ($w = Cr$) was comparable to the one involved in matrix-vector multiplication ($v = Ap$). On the other hand, the preconditioning quality has not deteriorated with the growth of the number of blocks (equal to the number of processors), and sometimes it has even enhanced. In numerical tests we typically observed parallel speedups varying from almost linear to superlinear.

As the test problems, we mainly used large-scale two- and three-dimensional Finite Element models in linear elasticity as arising in computational mechanics, with the order of system up to $8 \cdot 10^5$, number of nonzero elements in A up to $8 \cdot 10^7$, and the spectral condition number of order 10^{12}.

Acknowledgements

The authors are very grateful to Owe Axelsson for his interest in this work. This work has been supported by The National Science Foundation (NWO) grant B61-298, by INTAS-RFBR 0095-98 grant, by The Netherlands Technology Foundation (STW) grant NNS.4683 and by The National Science Foundation (NWO) grant 047.008.007.

References

1. Kaporin I.E. On preconditioned conjugate-gradient method for solving discrete analogs of differential problems. *Diff. Equat.* 1990; **26** (7): 897-906.
2. Kaporin I.E. Explicitly preconditioned conjugate gradient method for the solution of unsymmetric linear systems. *Int. J. Computer Math.* 1992; **40**: 169-187.
3. Kaporin I.E. New convergence results and preconditioning strategies for the conjugate gradient method. *Numerical Linear Algebra with Applications.* 1994; **1**: 179-210.
4. Kaporin I.E. High quality preconditioning of a general symmetric positive definite matrix based on its $U^TU + U^TR + R^TU$-decomposition. *Numerical Linear Algebra with Applications.* 1998; **5**: 483-509.
5. Karypis G, Kumar V. Multilevel *k*-way hypergraph partitioning, *Technical Report* 98-036, Dept. Comp. Sci. Engrg., Army HPC Research Center, Univ. of Minnesota, MN, 1998.
6. Kaporin IE, Konshin IN. Parallel solution of large sparse SPD linear systems based on overlapping domain decomposition. In *Parallel Computing Technologies*, Malyshkin V (ed), *Lecture Notes in Computer Science*, Vol. 1662, Springer-Verlag: Berlin - Heidelberg - New-York, 1999; 436-445.
7. Axelsson O, Kaporin I, Konshin I, Kucherov A, Neytcheva M, Polman B, Yeremin A. Comparison of algebraic solution methods on a set of benchmark problems in linear elasticity. *Tech. Report* of Department of Mathematics, University of Nijmegen, The Netherlands, 2000, 89 p.
8. Kaporin IE, Konshin IN. Parallel solution of symmetric positive definite systems based on decomposition into overlapping blocks. *J. Comp. Math. Math. Phys.* 2001; **41**(4): 481-493.
9. Kaporin I.E, Konshin I.N. A parallel block overlap preconditioning with inexact submatrix inversion for linear elasticity problems. *Numerical Linear Algebra with Applications.* 2002; **9**: 141-162.