Load Balancing of Parallel Block Overlapped Incomplete Cholesky Preconditioning

Igor Kaporin and Igor Konshin

Dorodnicyn Computing Center of Russian Academy of Sciences,
Vavilov str. 40, 119333 Moscow, Russia
{kaporin,konshin}@ccas.ru

Abstract. A modification of the second order Incomplete Cholesky (IC) factorization with controllable amount of fill-in is described and analyzed. This algorithm is applied to the construction of well balanced coarse-grain parallel preconditioning for the Conjugate Gradient (CG) iterative solution of linear systems with symmetric positive definite matrix. The efficiency of the resulting parallel algorithm is illustrated by a series of numerical experiments using large-scale ill-conditioned test matrices taken from the collection of the University of Florida.

Keywords: symmetric positive definite matrix, incomplete Cholesky factorization, conjugate gradient method, parallel preconditioning.

1 Introduction

An analysis of the parallel performance of parallel preconditioned CG solver described, e.g., in [1], [3], showed that the imbalance of the independent tasks at the stage of preconditioning application (due to the loosely controlled variations in the density of the IC factors of submatrices) is mainly responsible for the loss in the overall parallel efficiency.

A simple strategy considered below is to drop down the smallest entries in the most filled incomplete factors. Both matrix theory and numerical experiments show that a slight deterioration in the preconditioning quality should be more than compensated by the reduction of the time cost per iteration.

For the second order IC factorization [2] we propose to reassign the largest by magnitude entries from the error matrix to the IC factor. It may improve both the balancing and convergence rate.

2 Theoretical Analysis of the Post-filtering Techniques

Let us consider separately the cases of the (plain) IC factorization and the IC2 factorization [2] (with the structured error term). Hereafter, we will assume that A is symmetric positive definite matrix symmetrically scaled to the unit main diagonal.

V. Malyshkin (Ed.): PaCT 2009, LNCS 5698, pp. 304–315, 2009.

2.1 Plain IC Truncation

Let the IC factor U be obtained from the standard IC equation

$$A = U^T U - S \,, \tag{1}$$

where S is the error matrix, each element of which is $O(\tau)$ and τ is the truncation parameter, $0 < \tau \ll 1$. When one applies a Jennings–Malik type algorithm [5], the matrix S is symmetric nonnegative definite and therefore

$$\lambda(U^{-T} A U^{-1}) \leq 1 \,,$$

i.e., the eigenvalues of the preconditioned matrix is bounded by 1. At the same time, most eigenvalues of the preconditioned matrix are clustered around 1, which improves the numerical stability of the the corresponding preconditioned CG iterations [7].

Now we consider the splitting of U into the 'main' term and the 'error' term:

$$U = \tilde{U} + \tilde{R} \,,$$

where the entries of strictly upper triangular matrix R also do not exceed τ in magnitude.

Using the IC equation (1), one has then

$$\begin{aligned} A &= (\tilde{U} + \tilde{R})^T (\tilde{U} + \tilde{R}) - S \\ &= \tilde{U}^T \tilde{U} + (\tilde{U}^T \tilde{R} + \tilde{R}^T \tilde{U} + \tilde{R}^T \tilde{R} - S) \,. \end{aligned}$$

Hence,

$$\tilde{U}^T \tilde{U} = A + \tilde{S} \,,$$

where

$$\tilde{S} = S - \tilde{U}^T \tilde{R} - \tilde{R}^T \tilde{U} + \tilde{R}^T \tilde{R} \,,$$

i.e. $\tilde{U}$ is the exact IC of $A + \tilde{S}$, where the matrix $\tilde{S}$ is the perturbed error matrix. The latter also has $O(\tau)$ entries, but, in general, it is symmetric indefinite. Therefore, the norm $\|\tilde{U}^{-1}\|$ can be unbounded, and the corresponding preconditioning is not robust. Of course, the matrix S can be made positive definite by performing an IC factorization of the modified matrix, e.g., $A + \sigma I_n$ (cf. [6]), but the shift parameter should be as rough as $O(\tau)$ in order to guarantee the robustness. Therefore, the quality of such preconditioner is often insufficient, especially for ill-conditioned matrices A.

2.2 Truncation of IC2 Factor U

A special type *structured* error matrix arises in the IC2 factorization [2]

$$A = U^T U + U^T R + R^T U - S \,. \tag{2}$$

where R is a strictly upper triangular error matrix, each element of which is $O(\tau)$, S is a symmetric error matrix, each element of which is $O(\tau^2)$ and τ is the truncation parameter, $0 < \tau \ll 1$.

The following two facts (see [2]) are of key importance:

(a) for the same τ, both IC and IC2 factorizations have quite similar upper bounds on the fill-in for U, and

(b) on the contrary, the condition number of matrix A preconditioned by IC and IC2 are $O(\tau \mathrm{cond} A)$ and $O(\tau(\mathrm{cond} A)^{1/2})$, respectively. This explains the superior performance of IC2 preconditioning observed in its practical use (see, e.g., [2], [3]).

The following proposition can be readily deduced from results of [2].

Theorem 1. *For the preconditioned matrix*

$$M = U^{-T} A U^{-1}$$

defined according to (2), one has the estimate

$$\lambda(M) \le 1 + \gamma \tag{3}$$

whenever

$$R^T R \le \gamma S \tag{4}$$

holds.

Remark 1. A simple method which guarantees the validity of (4) is the use of an **a priori** diagonal shift of the order $O(\tau^2)$ for the original matrix A, cf. [2].

We now consider the splitting

$$U = \tilde{U} + \hat{R}\,, \tag{5}$$

where R includes certain amount of the smallest by magnitude nonzero elements of U, and estimate the quality of the preconditioning obtained by the use of truncated (or *post-filtered*) IC factor $\tilde{U}$.

Theorem 2. *For the preconditioned matrix*

$$\tilde{M} = \tilde{U}^{-T} A \tilde{U}^{-1}$$

defined according to (2) and (5), one has the estimate

$$\lambda(\tilde{M}) \le \frac{1+\gamma}{1-\hat{\gamma}-2\sqrt{\gamma\hat{\gamma}}} \tag{6}$$

whenever conditions (4) and

$$\hat{R}^T \hat{R} \le \hat{\gamma} S \tag{7}$$

with $\hat{\gamma} < (\sqrt{\gamma} + \sqrt{1+\gamma})^{-2}$ *hold.*

Remark 2. The other theoretical properties (e.g., the lower spectral bound) of the post-truncated IC2 remain essentially the same as for the original version presented in [2].

2.3 Truncation of IC2 Error Matrix R

Next we consider the splitting of the IC2 error matrix

$$R = \hat{R} + \tilde{R}\,, \tag{8}$$

where $\hat{R}$ includes certain amount of the *largest* by magnitude nonzero elements of R, and estimate the quality of the preconditioning obtained by the use of *augmented* IC2 factor

$$\tilde{U} = U + \hat{R}\,. \tag{9}$$

It appears that the only difference with the above case of IC2 factor truncation (see Theorem 2) is that the equation $\tilde{R} = R + \hat{R}$ is replaced by $\tilde{R} = R - \hat{R}$. Therefore, it can be readily shown that the corresponding estimate (6) holds under the same conditions (4) and (7).

Hence, in both cases one should take care about keeping the norm of the matrix $\hat{R}$ sufficiently small. However, when the truncation of error matrix R is performed, the norm $\|\hat{R}\|$ is not large since $\|R\|$ is bounded.

On the other hand, it makes sense to include into $\hat{R}$ the largest by the magnitude elements of R, thus making $\tilde{R}$ as small as possible. This well agrees with the following result related to the estimation of the K-condition number

$$\mathrm{K}(M) = \left(\frac{1}{n}\mathrm{trace}(M)\right)^n \Big/ \det M\,,$$

where M is the preconditioned matrix. (A detailed discussion of the relation of $\mathrm{K}(M)$ to convergence of the CG method can be found in [1], [2].)

Theorem 3. *For the preconditioned matrix*

$$\tilde{M} = \tilde{U}^{-T} A \tilde{U}^{-1}$$

defined according to (2) and (9), the following upper bound for the K*-condition number is valid,*

$$\mathrm{K}(\tilde{M}) \le (\det U)^2 / \det A\,, \tag{10}$$

whenever the condition

$$\tilde{R}^T \tilde{R} \le S + R^T R \tag{11}$$

holds.

Hence, if the matrix $\tilde{R}$ is sufficiently small by the norm, then the K-condition number of the preconditioned matrix has an upper bound which does not depend on this matrix.

3 Finding a Prescribed Amount of Smallest Elements in Array

According to (7), one should include in $\hat{R}$ the smallest entries of U in order to keep $\hat{\gamma}$ as small as possible in the case of post-filtering the IC2 factor. On the

contrary, if the IC2 error matrix is filtered, one may choose to find the largest entries of R.

An obvious approach is to sort the whole array U and include into the matrix $\hat{R}$ its smallest entries, which would cost $O(\mathrm{nz}(U)\log \mathrm{nz}(U))$ operations.

To reduce the cost of truncation we use hashing over the interval $[0,1]$ divided into n equal segments, where n is the dimension of the matrix A. After rehashing with the corrected interval boundaries we obtain the required threshold up to 1 element accuracy. The costs of such procedure is about $2n + 2\mathrm{nz}(U)$ operations.

4 A Description of Parallel IC2-Based Preconditioning

Let A be reordered and split in the same way as for the Block Jacobi preconditioning, i.e. the t-th diagonal block of the symmetrically reordered matrix has the dimension n_t and $n_1 + \ldots + n_p = n$. Here $t = 1, ..., p$, and p is the block dimension of A. For the t-th diagonal block, let us define the 'basic' index set as

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

where

$$k_{t-1} = n_1 + ... + n_{t-1} , \qquad k_0 = 0 , \quad k_p = n ,$$

and introduce the 'overlapping' index sets as

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

where

$$m_t \ge n_t , \qquad m_1 = n_1 .$$

For each t, the latter index set typically includes those indices not greater than k_t that are the most 'essentially' connected to the basic index set, e.g. in the sense of the sparse matrix graph adjacency relations. According to [1], [3], the Block Incomplete Inverse Cholesky (BIIC) preconditioner H is:

$$H = \sum_{t=1}^{p} V_t U_t^{-1} \begin{bmatrix} 0 & 0 \\ 0 & I_{n_t} \end{bmatrix} U_t^{-T} V_t^T ,$$

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

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

and each upper triangular matrix U_t is the (approximate) right Cholesky factor of the t-th 'extended' diagonal $m_t \times m_t$ submatrix $V_t^T A V_t$, that is,

$$V_t^T A V_t \approx U_t^T U_t , \quad t = 1, ..., p .$$

4.1 Block Splitting and Overlap

In our implementation, the matrix graph splitting is performed without any use of actual topology of underlying physical models. We use the public-domain graph partitioning package METIS [4] to divide the node set into p approximately equal 'subdomains' (or blocks).

The overlap is obtained using the sparsity structure of the qth degree of the coefficient matrix A^q. We refer to q as the overlap size parameter.

4.2 Load Balancing Strategies

Except of relatively small number of additions, an application of the parallel preconditioner BIIC2 (Block Incomplete Inverse Cholesky 2nd order) described in [3] is reduced to p independent lower and upper sparse triangular solves. Here p is the number of parallel processors available.

Even if a nearly equal partition of the matrix into overlapping blocks is used, the quantities $\mathrm{NZ}(U_t)$ may be unequal in general. In this case, at the iteration stage the triangular solves become imbalanced which may essentially deteriorate the overall efficiency.

The most obvious load balancing strategy consists in post-filtering of the incomplete factors. Here, one should use a somewhat finer truncation parameter at the factorization stage in order to maintain a sufficient preconditioning quality even with the post-filtered factors.

In what follows, we use the following balancing options:

'STD' denote the preconditioner obtained *as is* with no any post-processing;

'MIN' denotes the post-filtering of each matrix U_t in order to reduce all $\mathrm{NZ}(U_t)$ down to the minimum value over p blocks;

'MAX' denotes the augmentation of each U_t with the largest elements taken from the error matrix R_t in order to enlarge all $\mathrm{NZ}(U_t)$ up to their maximum value;

'AVR' denotes the combination of the latter two strategies in order to equalize $\mathrm{NZ}(U_t)$ near the arithmetic mean value.

5 Numerical Results

We have performed numerical experiments on MVS6000IM computer with Dual Core Intel Itanium 2 processors under RedHat Linux v.2.4.21-20 using MPICH for GM v.1.2.6.14b communication library for data transfer.

5.1 Test Problems and Solution Statistics

We have considered several most ill-conditioned matrices found in the University of Florida sparse matrix collection [8]. The matrix properties are presented in Table 1, where

'Matrix' is the matrix name;

n is the matrix size;

'NZ' is the number of nonzeros in matrix A, and

'Cond' is the estimated condition number.

In all experiments we take the right-hand side $b = Ax^*$, with $x^* \equiv 1$ as the exact solution and $x_0 \equiv 0$ as the initial guess.

In the construction of the preconditioner, the default drop tolerance threshold parameter τ is equal to 10^{-3}. The overlap size parameter q is equal to 10 in all cases.

The PCG iterations were performed until the accuracy $||Ax_k - b||/||b|| < \varepsilon = 10^{-8}$ was achieved.

In the tables below, we use the following notation:

p is the number of processors (blocks) used;

'Balanc' denotes the type of load balancing as indicated earlier in Subsection 4.2;

'Dens' is the preconditioner density $\mathrm{NZ}(U)/\mathrm{NZ}(U_A)$ with respect to the density of the upper triangle of matrix A;

'Imb' is the measure of imbalance in the sizes of preconditioner blocks defined as

$$\mathrm{Imb} = 1 - \frac{\sum_{i=1}^{p} \mathrm{NZ}(U_i)}{p \max_{1 \le i \le p} \mathrm{NZ}(U_i)} \; ;$$

'It' stands for number of iterations, and

'T_{wc}' stands for wall clock total solution time.

5.2 Test Results and Discussion

In the Tables 2–13, we present a comparison for the above mentioned four balancing strategies used with IC(τ), IC(τ^2), and IC2(τ, τ^2) preconditionings on $p = 1$ (serial version) and $p = 8$ processors, obtained for the test problems specified in Table 1.

Table 1. Matrix properties

Matrix	n	NZ	NZ/n	It(PJ)	Cond(A_s)
bcsstk25	15 439	252 241	16.3	3178	3.55×10^{6}
msc23052	23 052	1 154 814	50.0	19086	7.57×10^{7}
gridgena	48 962	512 084	10.4	3216	1.35×10^{5}
cvxbqp1	50 000	349 968	6.9	16	1.73×10^{1}
oilpan	73 752	3 597 188	48.7	18222	1.03×10^{8}
s3dkt3m2	90 449	3 753 461	41.4	40313	3.12×10^{10}
x104	108 384	10 167 624	93.8	64790	2.12×10^{9}
g2_circuit	150 102	726 674	4.8	881	2.34×10^{5}
BenElechi1	245 874	13 150 496	53.4	31501	1.84×10^{9}
msdoor	415 863	20 240 935	48.6	30081	1.95×10^{8}
af_1_k101	503 625	17 550 675	34.8	17321	4.43×10^{7}
parabolic_fem	525 825	3 674 625	6.9	1321	2.10×10^{5}

Table 2. Comparison of balancing strategies for 'bcsstk25' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	2.21	.00	404	3.67
IC(τ)	8	STD	4.07	.44	396	1.31
IC(τ)	8	MIN	1.80	.00	586	1.26
IC(τ^2)	1	STD	14.83	.00	17	2.29
IC(τ^2)	8	STD	29.91	.48	35	1.45
IC(τ^2)	8	MIN	8.14	.00	63	1.10
IC2(τ, τ^2)	1	STD	3.51	.00	50	2.20
IC2(τ, τ^2)	8	STD	5.74	.45	72	1.07
IC2(τ, τ^2)	8	MIN	2.18	.00	866	2.24
IC2(τ, τ^2)	8	AVR	4.71	.00	109	1.11
IC2(τ, τ^2)	8	MAX	9.24	.00	52	1.03

Table 3. Comparison of balancing strategies for 'msc23052' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	1.75	.00	1320	43.32
IC(τ)	8	STD	4.26	.44	1222	15.28
IC(τ)	8	MIN	1.55	.00	3143	17.54
IC(τ^2)	1	STD	12.89	.00	103	28.78
IC(τ^2)	8	STD	19.94	.57	157	15.92
IC(τ^2)	8	MIN	3.96	.00	1651	19.95
IC2(τ, τ^2)	1	STD	2.42	.00	343	25.89
IC2(τ, τ^2)	8	STD	5.59	.45	269	10.37
IC2(τ, τ^2)	8	MIN	1.87	.00	>10000	—
IC2(τ, τ^2)	8	AVR	5.36	.00	429	9.95
IC2(τ, τ^2)	8	MAX	8.44	.00	194	8.87

Table 4. Comparison of balancing strategies for 'gridgena' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	2.76	.00	198	5.89
IC(τ)	8	STD	3.26	.17	220	1.16
IC(τ)	8	MIN	2.57	.00	243	1.17
IC(τ^2)	1	STD	27.62	.00	11	7.80
IC(τ^2)	8	STD	15.05	.22	114	2.02
IC(τ^2)	8	MIN	10.27	.00	115	1.45
IC2(τ, τ^2)	1	STD	6.33	.00	59	7.22
IC2(τ, τ^2)	8	STD	5.88	.21	125	1.24
IC2(τ, τ^2)	8	MIN	4.42	.00	159	1.29
IC2(τ, τ^2)	8	AVR	5.88	.00	134	1.23
IC2(τ, τ^2)	8	MAX	7.34	.00	117	1.30

Table 5. Comparison of balancing strategies for 'cvxbqp1' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	3.95	.00	135	3.81
IC(τ)	8	STD	7.78	.44	189	2.38
IC(τ)	8	MIN	3.08	.00	1358	7.56
IC(τ^2)	1	STD	55.47	.00	19	23.01
IC(τ^2)	8	STD	52.06	.49	67	5.51
IC(τ^2)	8	MIN	10.32	.00	136	3.23
IC2(τ, τ^2)	1	STD	5.85	.00	48	19.53
IC2(τ, τ^2)	8	STD	10.38	.48	75	3.32
IC2(τ, τ^2)	8	MIN	3.44	.00	>10000	—
IC2(τ, τ^2)	8	AVR	9.17	.00	151	3.32
IC2(τ, τ^2)	8	MAX	16.13	.00	70	3.02

Table 6. Comparison of balancing strategies for 'oilpan' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	1.43	.00	1333	116.54
IC(τ)	8	STD	2.38	.31	1454	30.76
IC(τ)	8	MIN	1.27	.00	1566	21.03
IC(τ^2)	1	STD	10.32	.00	90	63.89
IC(τ^2)	8	STD	10.50	.33	98	12.71
IC(τ^2)	8	MIN	4.82	.00	118	9.24
IC2(τ, τ^2)	1	STD	1.82	.00	148	36.51
IC2(τ, τ^2)	8	STD	2.93	.31	110	7.17
IC2(τ, τ^2)	8	MIN	1.54	.00	6350	90.28
IC2(τ, τ^2)	8	AVR	2.93	.00	154	7.54
IC2(τ, τ^2)	8	MAX	3.93	.00	107	7.27

Table 7. Comparison of balancing strategies for 's3dkt3m2' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	1.77	.00	3728	403.38
IC(τ)	8	STD	2.31	.33	3740	82.29
IC(τ)	8	MIN	1.42	.00	3988	56.32
IC(τ^2)	1	STD	19.90	.00	192	199.48
IC(τ^2)	8	STD	12.33	.36	246	27.53
IC(τ^2)	8	MIN	6.82	.00	246	15.05
IC2(τ, τ^2)	1	STD	2.97	.00	247	79.84
IC2(τ, τ^2)	8	STD	3.61	.30	261	12.66
IC2(τ, τ^2)	8	MIN	2.17	.00	758	17.52
IC2(τ, τ^2)	8	AVR	3.61	.00	271	11.22
IC2(τ, τ^2)	8	MAX	5.06	.00	246	12.47

Table 8. Comparison of balancing strategies for 'x104' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	0.88	.00	5180	606.06
IC(τ)	8	STD	1.82	.40	6314	306.90
IC(τ)	8	MIN	0.75	.00	>10000	—
IC(τ^2)	1	STD	6.70	.00	292	247.29
IC(τ^2)	8	STD	13.39	.52	757	265.93
IC(τ^2)	8	MIN	3.55	.00	1084	123.10
IC2(τ, τ^2)	1	STD	1.12	.00	753	167.35
IC2(τ, τ^2)	8	STD	2.28	.40	941	101.21
IC2(τ, τ^2)	8	MIN	0.91	.00	5617	191.48
IC2(τ, τ^2)	8	AVR	2.23	.00	1143	93.49
IC2(τ, τ^2)	8	MAX	3.77	.00	795	95.53

Table 9. Comparison of balancing strategies for 'g2_circuit' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	4.33	.00	86	6.33
IC(τ)	8	STD	5.24	.10	124	1.82
IC(τ)	8	MIN	4.05	.00	134	1.77
IC(τ^2)	1	STD	54.87	.00	11	32.82
IC(τ^2)	8	STD	48.28	.12	26	4.42
IC(τ^2)	8	MIN	25.84	.00	25	3.80
IC2(τ, τ^2)	1	STD	6.69	.00	26	18.85
IC2(τ, τ^2)	8	STD	7.74	.10	38	2.84
IC2(τ, τ^2)	8	MIN	5.88	.00	54	2.82
IC2(τ, τ^2)	8	AVR	7.74	.00	37	2.68
IC2(τ, τ^2)	8	MAX	8.56	.00	36	2.69

Table 10. Comparison of balancing strategies for 'BenElechi1' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	1.79	.00	2450	985.59
IC(τ)	8	STD	2.18	.13	2535	169.36
IC(τ)	8	MIN	1.70	.00	2544	137.06
IC(τ^2)	1	STD	17.19	.00	152	558.59
IC(τ^2)	8	STD	14.03	.26	295	117.34
IC(τ^2)	8	MIN	9.30	.00	295	88.52
IC2(τ, τ^2)	1	STD	2.59	.00	276	295.10
IC2(τ, τ^2)	8	STD	3.04	.12	382	57.41
IC2(τ, τ^2)	8	MIN	2.32	.00	597	65.23
IC2(τ, τ^2)	8	AVR	3.04	.00	397	56.47
IC2(τ, τ^2)	8	MAX	3.47	.00	361	56.11

Table 11. Comparison of balancing strategies for 'msdoor' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	2.00	.00	1905	1275.95
IC(τ)	8	STD	2.50	.22	1972	247.80
IC(τ)	8	MIN	1.93	.00	1999	186.61
IC(τ^2)	1	STD	out	of	memory	—
IC(τ^2)	8	STD	17.40	.20	178	156.57
IC(τ^2)	8	MIN	10.77	.00	181	121.82
IC2(τ, τ^2)	1	STD	2.62	.00	198	495.71
IC2(τ, τ^2)	8	STD	3.23	.23	202	76.51
IC2(τ, τ^2)	8	MIN	2.42	.00	361	84.40
IC2(τ, τ^2)	8	AVR	3.24	.00	234	76.04
IC2(τ, τ^2)	8	MAX	4.23	.00	191	75.67

Table 12. Comparison of balancing strategies for 'af_1_k101' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	1.32	.00	908	405.86
IC(τ)	8	STD	1.45	.12	1023	72.73
IC(τ)	8	MIN	1.28	.00	1025	67.58
IC(τ^2)	1	STD	12.61	.00	60	254.40
IC(τ^2)	8	STD	9.74	.13	139	47.25
IC(τ^2)	8	MIN	6.91	.00	140	38.29
IC2(τ, τ^2)	1	STD	1.95	.00	163	170.64
IC2(τ, τ^2)	8	STD	2.01	.11	212	26.87
IC2(τ, τ^2)	8	MIN	1.77	.00	384	38.30
IC2(τ, τ^2)	8	AVR	2.01	.00	239	28.59
IC2(τ, τ^2)	8	MAX	2.25	.00	197	26.34

Table 13. Comparison of balancing strategies for 'parabolic_fem' problem

Method	p	Balanc	Dens	Imb	It	T_{wc}
IC(τ)	1	STD	3.24	.00	138	41.18
IC(τ)	8	STD	3.53	.15	214	9.86
IC(τ)	8	MIN	2.70	.00	231	8.83
IC(τ^2)	1	STD	42.60	.00	11	89.48
IC(τ^2)	8	STD	28.42	.13	70	17.84
IC(τ^2)	8	MIN	23.97	.00	70	16.02
IC2(τ, τ^2)	1	STD	5.29	.00	62	40.79
IC2(τ, τ^2)	8	STD	5.86	.12	97	7.72
IC2(τ, τ^2)	8	MIN	5.01	.00	108	7.67
IC2(τ, τ^2)	8	AVR	5.86	.00	100	7.63
IC2(τ, τ^2)	8	MAX	6.63	.00	93	7.57

It is seen that the most obvious load balancing based on the 'MIN' strategy often shows poor convergence. For instance, for the problem 'msc23052' the convergence is not achieved for 10000 iterations (see Table 3).

The sharp increase in the iteration number for many hard-to-solve problems (such as 'bcsstk25', 'msc23052', 'oilpan', 'cvxbqp1', 'x104') for BIIC2 preconditioning with 8 processors and 'MIN' balancing strategy is explained by the presence of very large $\lambda_{\max}(\tilde{M})$, as indicate the estimated bounds for this quantity. Note that typical values which guarantee the numerical stability of the PCG iterations are below 2. This destructive effect may be related to the violation of the upper bound on the norm of truncated parts of U_t, see Theorem 2 earlier.

On the other hand, the numerical results show that the proposed load balancing based on the 'MAX' strategy is sufficiently robust and efficient. In many cases, the use of 'MAX' balancing strategy resulted in a smaller total solution time as compared to the original (unbalanced) 'STD' version.

References

1. Kaporin, I.E.: New convergence results and preconditioning strategies for the conjugate gradient method. Numer. Linear Algebra Appls. 1, 179–210 (1994)
2. Kaporin, I.E.: High quality preconditionings of a general symmetric positive definite matrix based on its $U^TU + U^TR + R^TU$-decomposition. Numer. Lin. Alg. Appl. 5, 483–509 (1998)
3. Kaporin, I.E., Konshin, I.N.: A parallel block overlap preconditioning with inexact submatrix inversion for linear elasticity problems. Numer. Lin. Algebra Appl. 9, 141–162 (2002)
4. 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)
5. Jennings, A., Malik, G.M.: Partial elimination. J. Inst. Math. Appl. 20, 307–316 (1977)
6. Manteuffel, T.A.: An incomplete factorization technique for positive definite linear systems. Math. Comput. 34, 473–497 (1980)
7. Notay, Y.: On the convergence rate of the conjugate gradients in presence of rounding errors. Numer. Math. 65, 301–317 (1993)
8. University of Florida Sparse Matrix Collection, `http://www.cise.ufl.edu/research/sparse/matrices`