# Spectral analysis of parallel incomplete factorizations with implicit pseudo-overlap

M. Magolu monga Made[1],* and Henk A. van der Vorst[2]

[1] *Université Libre de Bruxelles, Service de Métrologie Nucléaire, C.P.I. 165/84 50, Av. F.D. Roosevelt, B-1050 Brussels, Belgium*
[2] *Utrecht University, Mathematical Institute, Mailbox 80.010 3508 Utrecht, The Netherlands*

## SUMMARY

Two general parallel incomplete factorization strategies are investigated. The techniques may be interpreted as generalized domain decomposition methods. In contrast to classical domain decomposition methods, adjacent subdomains exchange data during the construction of the incomplete factorization matrix, as well as during each local forward elimination and each local backward elimination involved in the application of the preconditioner. Local re-numberings of nodes are combined with suitable global fill-in strategy in a (successful) attempt to overcome the well-known trade-off between high parallelism (locality) and fast convergence (globality). From an algebraic viewpoint, our techniques may be implemented as global re-numbering strategies. Theoretical spectral analysis is provided, which displays that the convergence rate weakly depends on the number of subdomains. Numerical results performed on a 16-processor SGI Origin 2000 are reported, showing the efficiency of our parallel preconditionings.

KEY WORDS: Large sparse linear systems; incomplete factorizations; preconditioned conjugate gradient; multiprocessor computers

## 1. Introduction

Linear systems from boundary value problems like the diffusion equation can be solved by iterative methods. The speed of convergence depends very much on global properties (a local correction affects the whole solution), whereas for parallelism one wants to split the problem into smaller (almost) independent subproblems. These two requirements are in conflict [14, 43, 45]. A critical topical question in the use of incomplete factorization based

*Correspondence to: Université Libre de Bruxelles, Service de Métrologie Nucléaire, C.P.I. 165/84. 50, Av. F.D. Roosevelt, B-1050 Brussels. magolu@ulb.ac.be . http://homepages.ulb.ac.be/˜magolu

Contract/grant sponsor: European Commission. ESPRIT IV project; contract/grant number: 25009

preconditionings on parallel environments is how to overcome the above-mentioned trade-off between high level parallelism and rate of convergence [14, 15]. An answer to the above question requires to clearly identify why there is a trade-off. To this end, Doi and Lichnewsky [9, 10] relate this phenomenon to the number of *incompatible nodes* (any node $i$ which is connected to at least two nodes $j$ and $k$ *along the same direction (axis)*, such that $j \neq k$ and $i < j < k$) in the graph of the system matrix. They observed that the fewer the incompatible nodes, the faster the convergence, which may be explained by the fact that such nodes lead to (large) entries in the remainder matrix. Observe that, with red-black orderings combined with ILU(0), most of vertices are *incompatible nodes*. This gives rise to poor preconditioners, while higher fill levels ILU of red-black orderings, which reduce the number of incompatible nodes, are competitive preconditioners [14]. Recently in [7], Bridson and Tang used the concept of *reversed graph traversal* which is a straightforward generalization of the reverse Cuthill-McKee ordering. They showed that the closer an ordering is to some reversed graph traversal ordering the better its performance. Their analysis displayed that fill levels higher than zero are mandatory to make many well known orderings closer to a reversed graph traversal. In [11], Doi and Washio advocated to combine (multi-wavefront variants of) large-numbered multicolor orderings with a selective level-1 fill-in strategy. In the case of three-dimensional problems, an implicit block level-1 fill-in strategy has been applied in [26] to improve the performance of a zebra (or line red-black) like ordering suitable for parallel computations. In a series of papers (see [19] and references included), Hysom and Pothen proceeded into two steps. Interior unknowns are first eliminated within each subdomain. Interface unknowns are then handled through a coloring of the graph of subdomains. Their results show that high performance is achieved with fill levels higher than zero. It clearly emerges from the parallel ILU literature, roughly summarized here, that higher fill levels are necessary to restore some globality lost with parallel orderings.

In [28, 29], we have attempted to explain why, for the parallel orderings investigated in [24, 34, 18], the convergence deteriorates with an increasing number of subdomains. This led us to conclusions similar to those of Doi and Lichnewsky. Most of the nodes along (or next to) interfaces induce (rejected) fill-in entries which are responsible for the increase of the norm of the remainder matrix. We have also observed that including all level-1 fill-in entries, induced by the parallelization strategy, is not sufficiently effective in limiting the degradation of the convergence. This happens because (accepted) level-1 fill-in entries generate in turn (neglected) level-2 fill-in entries which are in general too big (in absolute value) to keep the norm of the remainder matrix comparable to the one for lexicographical ordering. Accepting all the fill-in entries (of any level), which are generated by the parallelization strategy, would annihilate the deterioration of the convergence rate, but unfortunately, it would seriously affect the parallelism. Numerical experiments reported in [28, 29] with two-dimensional domains show that a relative low fill level (around 4) is enough to obtain efficient parallel incomplete factorization preconditionings. The portion of the physical domain, where fill-in entries induced by the parallelism are accepted, is termed *pseudo-overlapping region*. Subdomain $\mathcal{P}_s$ is *pseudo-overlapped* by subdomain $\mathcal{P}_t$ if, during the construction of the incomplete factorization matrix factor(s) and during the forward substitution of the preconditioning solve, $\mathcal{P}_t$ has to send information to $\mathcal{P}_s$. An explicit knowledge of the *pseudo-overlapping region* enables for an easy overlap of computation with communication. Processor $\mathcal{P}_t$ accumulates and sends information to $\mathcal{P}_s$ as soon as all the entries in the *pseudo-overlapping region* are updated; the actual receipt by $\mathcal{P}_s$ occurs only at due time. For recent surveys of parallelization techniques, we refer to [12, 15].

In the present paper, we investigate two parallel incomplete factorization strategies in which adjacent subdomains implicitly pseudo-overlap each other. In doing so, we obtain a simple generalization of the so-called vdv four-processor orderings [44, 14]. The new techniques combine local re-numberings of nodes with properly chosen fill level, possibly with a dropping test to exclude inappropriate connections. Our ordering can be interpreted as (a generalization of) a reverse variant of an ordering discussed in [18]. The rather poor performance observed in [18] accounts for the fact that, for the ordering proposed there, all the gridpoints on the interfaces between subdomains are *incompatible nodes*. The ordering strategy, introduced in [34], combines both above-mentioned orderings (nonreverse and reverse) in a sophisticated way that results in an implicit minimization of the remainder matrix norm, provided that the grid is well structured and the fill level is set to zero. In contrast to [18] and [34], we include high fill levels. Without this, incomplete factorizations cannot compete with the additive-Schwarz domain decomposition methods, see, e.g., [18, Section 6], [39] or [28, 29]. Our techniques, which could be interpreted as generalizations of domain decomposition methods, may also be implemented as global re-numbering strategies. Adjacent subdomains have to exchange data during the construction of the incomplete factorization matrix, as well as during each forward elimination and each backward elimination involved in the application of the preconditioner. This differentiates our techniques from classical (additive Schwarz and Schur complement based) domain decomposition methods, where both local forward and backward substitutions apply without any communication between adjacent subdomains. We establish theoretical relations between the spectral condition number and the number of subdomains. For a given matrix $A$ preconditioned by some (parallel) incomplete factorization matrix $B$, we obtain upper spectral (condition number) bound of the form:

$$\kappa(\mathbf{B^{-1}A}) \le \alpha\mathbf{h^{-2}} + (\beta\mathbf{p} + \gamma)\,\mathbf{h^{-1}} + \eta - \mu\mathbf{p}, \tag{1.1}$$

where $h$ stands for the mesh size parameter, $p$ denotes the number of subdomains, $\alpha$, $\beta$, $\gamma$, $\eta$ and $\mu$ denote some parameters independent of both $h$ and $p$. This clearly displays that the convergence rate of our parallel ILU is weakly dependent on the number of subdomains.

Our exposition is organized as follows. In Section 2 we collect some terminology and notation. Section 3 contains the main algorithm tools: the preconditioned conjugate gradient (PCG), and a general version of standard incomplete factorization. In Section 4, we describe our two parallelization approaches. Spectral analysis is carried out in Section 5. This results in insight in the convergence to be expected for our parallel orderings. Results of numerical experiments, for two-dimensional problems on a 16-processor SGI Origin 2000, are reported in Section 6. Conclusions are drawn in Section 7.

## 2. Terminology and notation

### *2.1. Stieltjes matrices*

A real square matrix $A$ is called a *Stieltjes matrix* (or equivalently, a *symmetric M-matrix*) if it is symmetric positive definite and none of its offdiagonal entries is positive [40, 6].

### *2.2. Connected gridpoints*

Two gridpoints $i$ and $j$ are *connected* with respect to the graph of $A$, if $a_{i,j} \neq 0$ or $a_{j,i} \neq 0$.

### 2.3. *Miscellaneous symbols*

We will use the following notation:

| | | |
|---|---|---|
| $A \in \mathbb{R}^{n\times n}$ | : | its elements are denoted with $a_{i,j}$ |
| $A^t$ | : | the transpose of $A$ |
| $\lambda_{\min}(A)$ | : | the smallest eigenvalue of $A$ |
| $\lambda_{\max}(A)$ | : | the largest eigenvalue of $A$ |
| $\lambda_i(A)$ | : | the $ith$ eigenvalue of $A$ <br> $\lambda_{\min}(A) = \lambda_1(A) \le \lambda_2(A) \le \cdots \le \lambda_n(A) = \lambda_{\max}(A)$ |
| $\kappa(A)$ | : | $\lambda_{max}(A)/\lambda_{min}(A)$ (the spectral condition number) |
| diag$(A)$ | : | the pointwise diagonal matrix whose diagonal entries coincide with those of $A$ |
| $\mathbf{e}$ | : | the vector with all components equal to 1 |

### 2.4. *$LPL^t$-factorization*

By the *$LPL^t$ factorization* of a nonsingular Stieltjes matrix $S$, we understand the (exact) factorization $S = L_s P_s L_s^t$, where $P_s$ is a diagonal matrix and $L_s$ is a lower triangular matrix with diag$(L_s) = I$.

## 3. Preliminaries

Our approach is more easily explained if there is some underlying grid. In order to make the method more understandable, and for analysis purposes, we consider the self-adjoint second order two-dimensional (elliptic) boundary value problem:

$$\begin{aligned} -\left(\alpha\, u_x\right)_x - \left(\beta\, u_y\right)_y &= f(x,y) && \text{in } \Omega = (0,1)\times(0,1) \\ u &= 0 && \text{on } , \\ \frac{\partial u}{\partial n} &= 0 && \text{on } \partial\Omega\backslash\, , \end{aligned} \tag{3.1}$$

where n is the outer normal to the boundary $\partial\Omega$ of $\Omega$. , denotes a nonempty portion of $\partial\Omega$. The coefficients $\alpha$ and $\beta$ are positive, bounded and piecewise constant. The PDE (3.1) is discretized over a uniform rectangular grid of mesh size $h$ in both directions with the five-point box integration scheme [32, 40]. We use the lexicographical ordering in the $(x, y)$-plane to number the unknowns. The resulting system matrix $A$ is a nonsingular block-tridiagonal, irreducibly diagonally dominant, Stieltjes matrix. The preconditioned conjugate gradient (PCG) method, which is given in Fig. 1, is a common method for such systems (see, e.g., [3, 17]). As a preconditioner $B$, we use the standard incomplete factorization with fill level ([30, 31]). The algorithm for the construction of $B$ ($LPL^t$ version) is shown in Fig. 2. The symbol $\mathcal{D}$ refers to the set of discarded fill-in entries:

$$\mathcal{D} = \{\, (k,i) \mid lev(l_{k,i}) > \ell \,\} \ ,$$

where $\ell$ denotes a user specified maximal fill level. With respect to the notation of Fig. 2, $lev(l_{k,i})$ is defined as follows:

Initialization: $lev(l_{k,i}) := \begin{cases} 0 & \text{if} \ \ l_{k,i} \neq 0 \ \ \text{or} \ \ k = i \\ \infty & \text{otherwise} \end{cases}$

Factorization: $lev(l_{k,i}) := \min\{\, lev(l_{k,i})\,,\, lev(l_{i,j}) + lev(l_{k,j}) + 1 \,\}$ .

```
 1. $r^{(0)} := b - A\,u^{(0)}$
 2. For $i = 1, 2, \ldots$ (until convergence)
 3.         Solve $w^{(i)}$ from
            $Bw^{(i)} := r^{(i)}$
 4.         $\gamma_i := (w^{(i)}, r^{(i)})$
 5.         $\beta_i := \begin{cases} 0 & \text{if} \quad i = 0 \\ \frac{\gamma_i}{\gamma_{i-1}} & \text{otherwise} \end{cases}$
 6.         $p^{(i)} := w^{(i)} + \beta_i\, p^{(i-1)}$
 7.         $w^{(i)} := A\, p^{(i)}$
 8.         $\alpha_i := \frac{\gamma_i}{(p^{(i)}, w^{(i)})}$
 9.         $u^{(i+1)} := u^{(i)} + \alpha_i\, p^{(i)}$
10.         $r^{(i+1)} := r^{(i)} - \alpha_i\, w^{(i)}$
```

Figure 1. Preconditioned conjugate gradient method. $B$ represents the preconditioning matrix.

## 4. ParIC($\ell$): A Family of Parallel Incomplete Factorizations

We consider only interfaces that coincide with the grid. In finite element terminology: each small square of the rectangular grid is assimilated to a finite element. The global matrix is partitioned into submatrices as in classical domain decomposition methods with minimal overlap [39]: interfaces between subdomains are made up of points and lines for 2D PDE matrices; points, lines, and planes for 3D PDE matrices. We further assume that the user wants to perform the discretization in parallel too, and that he has a complete freedom on how to decompose the physical domain into a number of subdomains, for instance by means of some automatic graph partitioning algorithm applied to the graph of finite elements (the dual graph) [13, 27].

### *4.1. Subdividing the gridpoints into classes*

In each subdomain, the (local) unknowns are re-numbered class by class, consecutively, as follows:

```
Compute $P$ and $L$  ($B = LPL^t$  with  $diag(L) = I$)

Initialization phase
$p_{i,i} := a_{i,i}$ ,   $i = 1, 2, \cdots, n$
$l_{i,j} := a_{i,j}$ ,   $i = 2, 3, \cdots, n$ ,   $j = 1, 2, \cdots, i-1$

Incomplete factorization process
    do $j = 1, 2, \cdots, n-1$
        do $i = j+1, j+2, \cdots, n$
            $p_{i,i} := p_{i,i} - \frac{l_{i,j}^2}{p_{j,j}}$
            $l_{i,j} := \frac{l_{i,j}}{p_{j,j}}$
            do $k = i+1, i+2, \cdots, n$
                if  $(k,i) \notin \mathcal{D}$  $l_{k,i} := l_{k,i} - l_{i,j}\, l_{k,j}$
            end do
        end do
    end do
```

Figure 2. Standard incomplete factorization (IC).

1. class 1: all interior gridpoints are numbered first;
2. class 2: next follow all gridpoints, if any, that belong to two subdomains (see Figs. 3 and 4);
3. class 3: next follow all gridpoints, if any, that belong to three subdomains (see Fig. 4);
4. etc ...

During the construction of the preconditioning matrix, and at each forward elimination, the computation and the exchange of data is performed, class by class, as follows:

1. compute class 1 gridpoints;
2. exchange data for class 2 gridpoints updates; compute class 2 gridpoints;
3. exchange data for class 3 gridpoints updates; compute class 3 gridpoints;
4. exchange data for class 4 gridpoints updates; etc ...

The ordering of computation and communication has to be reversed at each backward substitution. Steps that involve an empty class are assumed to be skipped. At gridpoints that are shared by two or more subdomains, each involved subdomain must hold the same

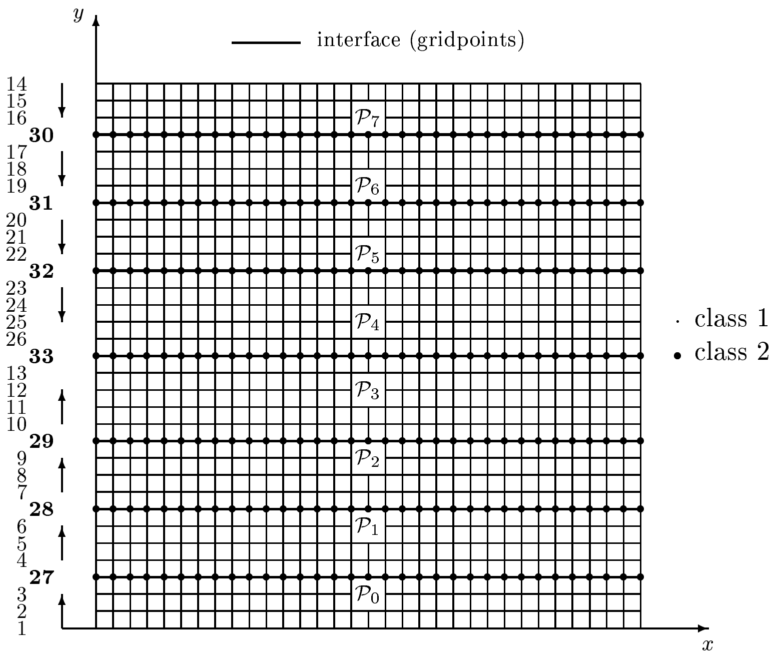

Figure 3. Decomposition of the grid into $1 \times 8$ (stripes). Arrows indicate the progressing direction of the line numbering per subdomain. Numbers along the $y$-axis give an example of a global (line) numbering of unknowns.

value (up to round-off errors) at the end of the incomplete factorization process, and during the preconditioning steps. To this end, connections between gridpoints of the same class, but that belong to two different interfaces, should be avoided. It may happen that the connection to be dropped corresponds to some entry $a_{i,j}$ of the original system matrix. In such a situation, the dropped value may be added to the diagonal entries $a_{i,i}$ and $a_{j,j}$. This technique, which is known as *diagonal compensation* [1], preserves the rowsum of the system matrix. In the case of the relaxed variants of IC introduced in [2], all the dropped values, possibly weighted, will be automatically added to the diagonal entries.

Figs. 3 and 4 show two examples where the physical domain is partitioned into $1\times 8$ (stripes) and $2\times 4$ subdomains, respectively.

**Definition 1.** *A standard IC($\ell$) combined with the above parallelization strategy will be referred to as ParIC($\ell$).*

This strategy defines, through the fill-ins, implicitly a pseudo-overlap. The pseudo-overlap width will be in general variable but bounded. Its size, which depends on the local numbering, cannot be larger than $\ell + 1$.

*Remarks:* Though the way we have chosen to explain our approach follows domain decomposition terminology, our method is different from and more efficient than classical

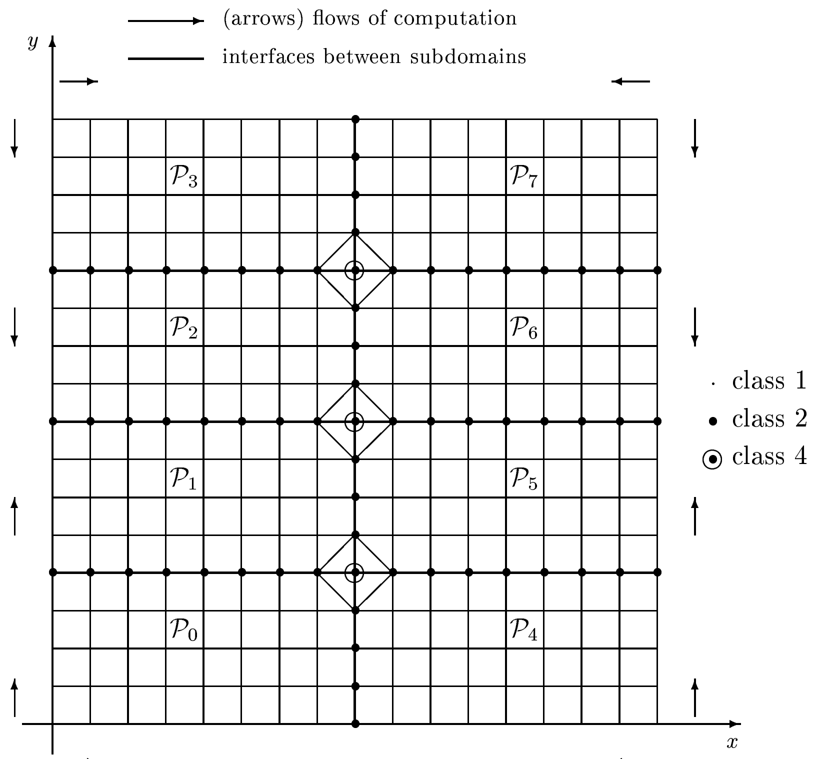

Figure 4. Decomposition of the grid into $2\times4$ subdomains. Oblique lines specify level-1 fill-in entries that are prohibited.

domain decomposition techniques. Numerical evidence may be found in [28, 29]. Moreover, our parallelization method described above may also be implemented as a purely algebraic global re-numbering strategy. This requires the following steps.

1. The graph of the global matrix is partitioned into submatrices by means of some algorithm which computes a *vertex separator*, say any (preferably small) set of vertices whose removal splits the remaining vertices in half or more (approximately) equal parts (see, e.g., [13, 14, 27]).
2. All the vertices outside the vertex separator (interior vertices) are (re)numbered first. Then follow the separator vertices, that are classified and numbered in increasing order according to the number of subdomains they are adjacent to.
3. Any connection between two separator vertices that are adjacent to a same number of subdomains is cancelled whenever the vertices involved have not the same list of adjacent subdomains.

Fig. 3 provides us with such a global reordering strategy. The vertex separator is made up of all the vertices along lines 27 to 33.

### *4.2. Subdividing the gridpoints into classes and subclasses*

The approach described in Section 4.1 is quite general. It may be used to handle any unstructured finite element (or finite volume) grid. A way to attempt to improve the performance consists of avoiding to drop as many low level fill-in entries as possible during the incomplete factorization process. This could be achieved by distinguishing between gridpoints of the same class. In the case of the partitioning described in Fig. 4, one may collect class 2 gridpoints into two subclasses according to whether a gridpoint belongs to a *vertical* (class 2a) or a *horizontal* (class 2b) interface (see Fig. 5). It is evident that splitting classes into subclasses is easy only for well-structured partitionings.

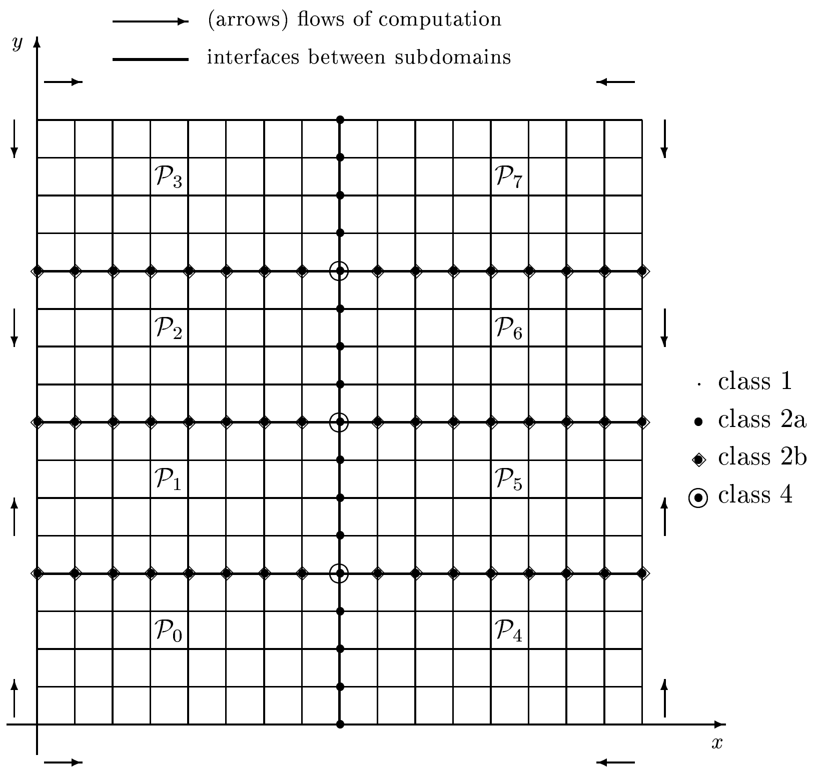

Figure 5. Decomposition of the grid into 2×4 subdomains, and partition into classes and subclasses.

Now, the computation and communication can be executed as follows:

1. compute class 1 gridpoints;
2. exchange data for class 2a gridpoints updates; compute class 2a gridpoints;
3. exchange data for class 2b gridpoints updates; compute class 2b gridpoints;
4. exchange data for class 4 gridpoints updates; compute class 4 gridpoints.

**Definition 2.** *The above variant of parallel incomplete factorization will be referred to as ParIC*($\ell$).*

## 5. Spectral Analysis

The rate of convergence of the PCG method depends on the eigenvalue distribution of the preconditioned system $B^{-1}A$. An upper bound for the number of iterations necessary to achieve a given accuracy is bounded above by the square root of the spectral condition number $\kappa(B^{-1}A)$ (see, e.g., [1, 17, 33, 36, 37, 38, 46]). Our aim in this section is to provide simple estimates for the extreme eigenvalues of $B^{-1}A$. We consider the remainder matrix $R = B - A$, and we define the diagonal matrix $\mathcal{X}$ by

$$B\mathbf{e} = A\mathbf{e} + \mathcal{X}\mathbf{e} \ . \tag{5.1}$$

The matrix $\mathcal{X}$ denotes the diagonal (perturbation) matrix; its diagonal entries are the rowsums of $R$. Therefore, any incomplete factorization preconditioning matrix may be interpreted as a *perturbed modified incomplete factorization*, where the perturbation matrix is defined as the rowsum of the remainder matrix. Due to this observation [20, Section 3], all theoretical results developed for (perturbed) modified incomplete factorizations (see, e.g., [4, 20, 21, 22, 23, 25]) transfer to the standard IC. Of course, only those results that yield good estimates are of interest. For instance, upper bounds of $\mathcal{O}(h^{-1})$ for $\lambda_{\max}(B^{-1}A)$, as in MIC type preconditioners, are not useful for general IC.

**Lemma 1.** [22, Theorem 4.3] *Assume that the preconditioning matrix $B = LPL^t$ is the incomplete factorization of $A$ with fill level zero. Define:*

$$\gamma_{i,i} = \begin{cases} \frac{p_{i,i}}{2p_{i,i}-a_{i,i}} & \text{if } \ i \notin \mathcal{M} \ , \\ \frac{a_{i,i}}{p_{i,i}} & \text{if } \ i \in \mathcal{M} \ , \end{cases} \tag{5.2}$$

*where*

$$\mathcal{M} = \{ \, i \, ; \, 1 \le i \le n \, , \, \mathcal{S}_i = \emptyset \, \} \ \ , \tag{5.3}$$

*with*

$$\mathcal{S}_i = \{ \, j \, ; \, i < j \le n \, , \, a_{i,j} \ne 0 \, \} \ \ . \tag{5.4}$$

*If, for all $i \notin \mathcal{M}$, $2p_{i,i} - a_{i,i}$ is positive then*

$$\lambda_{\max}(B^{-1}A) \le \max_{1 \le i \le n} \gamma_{i,i} \ \ . \tag{5.5}$$

**Lemma 2.** *Set $D = \mathrm{diag}(A)$. There exist real numbers $\epsilon_{h,i}$, which depend upon the mesh size parameter $h$ and $i$, $i \ll n$, such that*

$$\lambda_i(B^{-1}A) \ge \left( 1 + (1 + \epsilon_{h,i}) \, \frac{(\mathbf{e}, R\mathbf{e})}{(\mathbf{e}, D\mathbf{e})} \, \frac{1}{\lambda_i(D^{-1}A)} \right)^{-1} \ \ . \tag{5.6}$$

*Proof* The result follows from [25, Corollary 4.2], which gives

$$\lambda_i(B^{-1}A) \ge \left( 1 + \frac{\lambda_{\max}(D^{-1}\mathcal{X})}{\lambda_i(D^{-1}A)} \right)^{-1} \ .\Box \tag{5.7}$$

*Remark:* The lower bound (5.6) holds for any fill level. The parameters $\epsilon_{h,i}$ are not easy to compute. We shall use $\epsilon_{h,i} = 0$, which in general gives rise to accurate estimates (see, e.g., [4] for $i = 1$, and [25, Section 5] for $i \ll n$).

In order to give more specific spectral bounds, we shall make use of asymptotic results (limit matrix analysis), as in [41, 42, 8, 16, 46]. For simplicity, our study will be restricted to the case of fill level zero, applied to the PDE (3.1) with $\alpha = \beta = \mathbf{1}$, and $\Gamma = \partial\Omega$. There holds ([40]):

$$\lambda_{\min}(D^{-1}A) = 1 - \cos \pi h = \frac{\pi^2 h^2}{2} + \mathcal{O}(h^4) \ . \tag{5.8}$$

Furthermore, we have that

$$(\mathbf{e}, R\mathbf{e}) \ = \ \sum_{i=1}^{n} \sum_{j=1}^{n} r_{i,j} \ , \tag{5.9}$$

$$(\mathbf{e}, D\mathbf{e}) \ = \ \sum_{i=1}^{n} a_{i,i} = 4n = \frac{4(1-2h+h^2)}{h^2} \ . \tag{5.10}$$

We will further assume that the number of subdomains $p$ is even.

*5.1. Stripes (or $1 \times p$) partitionings*

We will compute the limit value of the diagonal entries $p_{i,i}$ of $P$ by means of the rules given in Fig. 2. Four cases may be distinguished:

**Case 1.** *$p_{i,i} \to \tilde{\rho}$ along the starting lines (lines 1,4,7,10,14,17,20,and 23, in Fig. 3), where $\tilde{\rho} \approx 3.7320508$ is the larger root of the quadratic equation $\tilde{\rho} = 4 - \frac{1}{\tilde{\rho}}$.*

**Case 2.** *$p_{i,i} \to \rho$ at the remaining interior (class 1) gridpoints, where $\rho \approx 3.4142136$ is the larger root of $\rho = 4 - \frac{2}{\rho}$.*

**Case 3.** *$p_{i,i} \to \hat{\rho}$ along the middle line (line 33 in Fig. 3), where $\hat{\rho} \approx 3.0906579$ is the larger root of $\hat{\rho} = 4 - \frac{1}{\hat{\rho}} - \frac{2}{\rho}$.*

**Case 4.** *$p_{i,i} \to \bar{\rho}$ at the remaining interface (class 2) gridpoints, where $\bar{\rho} \approx 3.1184895$ is the larger root of $\bar{\rho} = 4 - \frac{1}{\bar{\rho}} - \frac{1}{\rho} - \frac{1}{\tilde{\rho}}$.*

Here, $\mathcal{M}$ is the set of all gridpoints $i$ such that $i$ is the last (numbered) gridpoint of an interface (lines 27–33 in Fig. 3) between two adjacent subdomains. Therefore, due to Lemma 1, it follows that

$$\lambda_{\max}(B^{-1}A) \lesssim \begin{cases} 1.20711 & \text{if } \ p = 1 \ , \\ 1.41698 & \text{if } \ p \geq 2 \ . \end{cases} \tag{5.11}$$

Turning to the lower spectral bound, we first point out that all the rejected level-1 fill-in entries, which correspond to the nonzero entries of the remainder matrix $R$, are of the form $\frac{1}{p_{i,i}}$. Taking limit values ($i \to \infty$), it follows that:

$$(\mathbf{e}, R\mathbf{e}) \approx \begin{cases} 2(n_x - 1)\left(\frac{1}{\tilde{\rho}} + (n_y - 2)\frac{1}{\rho}\right) & \text{if } \ p = 1 \ , \\ 2(n_x - 1)\left(\frac{3p-4}{\bar{\rho}} + (n_y - 2p + 1)\frac{1}{\rho}\right) & \text{if } \ p \geq 2 \ . \end{cases} \tag{5.12}$$

Hence, by Lemma 2, we obtain the following lower spectral bounds:

$$\lambda_{\min}(B^{-1}A) \gtrsim \begin{cases} \left(1 + \frac{(n_x-1)\left(\frac{1}{\tilde\rho}+(n_y-2)\frac{1}{\rho}\right)}{2n_xn_y(1-\cos\frac{\pi}{n_x+1})}\right)^{-1} & \text{if } p=1 \ , \\ \left(1 + \frac{(n_x-1)\left(\frac{3p-4}{\tilde\rho}+(n_y-2p+1)\frac{1}{\rho}\right)}{2n_xn_y(1-\cos\frac{\pi}{n_x+1})}\right)^{-1} & \text{if } p\geq 2 \ . \end{cases} \tag{5.13}$$

With some elementary algebraic calculations, taking into account that $h = \frac{1}{n_x+1} = \frac{1}{n_y+1}$ and $1-\cos x \approx \frac{x^2}{2}$, and neglecting $\mathcal{O}(h^3)$-terms, one obtains the following estimates:

$$\lambda_{\min}(B^{-1}A) \gtrsim \begin{cases} \pi^2\rho h^2\{1 - (5-\frac{\rho}{\tilde\rho})h \\ \qquad + (\pi^2\rho+6-2\frac{\rho}{\tilde\rho})h^2\}^{-1} & \text{if } p=1 \ , \\ \pi^2\rho h^2\{1 + [(3p-4)\frac{\rho}{\tilde\rho}-2(p+1)]h \\ \qquad + [\rho\pi^2-2(3p-4)\frac{\rho}{\tilde\rho}+4p]h^2\}^{-1} & \text{if } p\geq 2 \ . \end{cases} \tag{5.14}$$

The symbol "$\gtrsim$" denotes that we have neglected $\mathcal{O}(h^3)$-terms. Substituting $\rho \approx 3.4142136$ (case 2) and $\tilde\rho \approx 3.7320508$ (case 1), one gets

$$\lambda_{\min}(B^{-1}A) \gtrsim \begin{cases} \frac{33.697}{1-4.085\,h+37.867\,h^2}\,h^2 & \text{if } p=1 \ , \\ \frac{33.697}{1+(0.745\,p-5.659)\,h+(41.016-1.489\,p)h^2}\,h^2 & \text{if } p\geq 2 \ . \end{cases} \tag{5.15}$$

Therefore

$$\kappa(B^{-1}A) \lesssim \begin{cases} 0.036\,h^{-2} - 0.15\,h^{-1} + 1.36 & \text{if } p=1 \ , \\ 0.042\,h^{-2} + (0.03\,p - 0.24)\ h^{-1} + 1.72 - 0.06\,p & \text{if } p\geq 2 \ . \end{cases} \tag{5.16}$$

Note that for these bounds, we have ignored the initial convergence behavior of the $p_{i,i}$. This has only a small effect on the eventual results (see [46]).

## *5.2. $2 \times m$-partitionings*

There is no difference between ParIC(0) and ParIC*(0). The computations are similar to the partitioning by stripes. The letter $m$ denotes the number of subdomains along the $y$-direction. The total number of subdomains is $p = 2m$.

**Case 1.** $p_{i,i} \to \tilde\rho \approx 3.7320508$ *at the first* $\frac{n_x}{2}$ *(if* $n_x$ *is odd) or* $\frac{n_x}{2}-1$ *(if* $n_x$ *is even) interior gridpoints of each subdomain.*

**Case 2.** $p_{i,i} \to \rho \approx 3.4142136$ *at the remaining class 1 gridpoints.*

**Case 3.** $p_{i,i} \to \hat\rho \approx 3.0906579$ *along the middle (horizontal line and vertical line) class 2 gridpoints.*

**Case 4.** $p_{i,i} \to \bar\rho \approx 3.1184895$ *at the remaining class 2 gridpoints.*

**Case 5.** $p_{i,i} = \ddot\rho \approx 4 - \frac{4}{\hat\rho} \approx 2.7057772$ *at the central class 4 gridpoint.*

**Case 6.** $p_{i,i} = \dot\rho \approx 4 - \frac{2}{\hat\rho} - \frac{2}{\bar\rho} \approx 2.7115524$ *at the remaining class 4 gridpoints.*

Here $\mathcal{M}$ is the set of all class 4 gridpoints (see Figs. 4 and 5). This implies with Lemma 1 that:

$$\lambda_{\max}(B^{-1}A) \lesssim \begin{cases} 1.41698 & \text{if } p=2 \ , \\ 1.47832 & \text{if } p>2 \ . \end{cases} \tag{5.17}$$

Now, one has that,

$$(\mathbf{e}, R\mathbf{e}) \approx \begin{cases} 2(n_x-1)\left(\frac{1}{\tilde{\rho}} + (n_y-2)\frac{1}{\rho}\right) & \text{if } p=2 \ , \\ 2(n_x-1)\left(\frac{3p-8}{2\tilde{\rho}} + (n_y-p+1)\frac{1}{\rho}\right) + \frac{p-4}{\rho_0} & \text{if } p>2 \ , \end{cases} \tag{5.18}$$

where $\rho_0 = 4 - \frac{2}{\tilde{\rho}} \approx 3.4641016$, whence it follows that,

$$\lambda_{\min}(B^{-1}A) \gtrsim \begin{cases} \left(1 + \frac{(n_x-1)\left(\frac{1}{\tilde{\rho}}+(n_y-2)\frac{1}{\rho}\right)}{2n_xn_y(1-\cos\frac{\pi}{n_x+1})}\right)^{-1} & \text{if } p=2 \ , \\ \left(1 + \frac{(n_x-1)\left(\frac{3p-8}{2\tilde{\rho}}+(n_y-p+1)\frac{1}{\rho}\right)+\frac{p-4}{\rho_0}}{2n_xn_y(1-\cos\frac{\pi}{n_x+1})}\right)^{-1} & \text{if } p>2 \ . \end{cases} \tag{5.19}$$

By dropping $\mathcal{O}(h^3)$-terms one gets

$$\lambda_{\min}(B^{-1}A) \gtrsim \begin{cases} \pi^2\rho h^2\{1 - (5-\frac{\rho}{\tilde{\rho}})h \\ \qquad + (\pi^2\rho + 6 - 2\frac{\rho}{\tilde{\rho}})h^2\}^{-1} & \text{if } p=2 \ , \\ \pi^2\rho h^2\{1 + [(\frac{3\rho}{2\tilde{\rho}}-1)p - 2(1+\frac{2\rho}{\tilde{\rho}})]h \\ \qquad + [(2-\rho(\frac{3}{\tilde{\rho}}-\frac{1}{2\rho_0}))p + \rho(\pi^2+\frac{8}{\tilde{\rho}}-\frac{2}{\rho_0})]h^2\}^{-1} & \text{if } p>2 \ , \end{cases} \tag{5.20}$$

or, approximately,

$$\lambda_{\min}(B^{-1}A) \gtrsim \begin{cases} \frac{33.697}{1-4.085\,h+37.867\,h^2}\,h^2 & \text{if } p=2 \ , \\ \frac{33.697}{1+(0.372\,p-5.659)\,h+(39.044-0.252\,p)h^2}\,h^2 & \text{if } p>2 \ . \end{cases} \tag{5.21}$$

This, together with (5.17), and (5.16) for the case of $p=1$, gives

$$\kappa(B^{-1}A) \lesssim \begin{cases} 0.036\,h^{-2} - 0.15\,h^{-1} + 1.36 & \text{if } p=1 \ , \\ 0.042\,h^{-2} - 0.17\,h^{-1} + 1.59 & \text{if } p=2 \ , \\ 0.044\,h^{-2} + (0.016\,p - 0.248)\ h^{-1} + 1.713 - 0.011\,p & \text{if } p>2 \ . \end{cases} \tag{5.22}$$

It emerges from our analysis that, *the convergence strongly deteriorates only if the number of subdomains p is such that $p \geq \mathcal{O}(h^{-1}) = \mathcal{O}(\sqrt{n})$. This is clearly a very bad choice: the overall communication time will (strongly) dominate the overall computation time.* In Figs. 6 and 7, we give the evolution of the eigenvalue lower bound estimates (5.15) and (5.21), which we denote by low(h,p), and the evolution of the spectral condition number estimates (5.16) and (5.22), which we denote by $k(h,p)$, in function of the number of subdomains $p$ ($p \leq 32$). Three values of the mesh size $h$ have been considered: $1/h = 481, 721, 961$. As the number

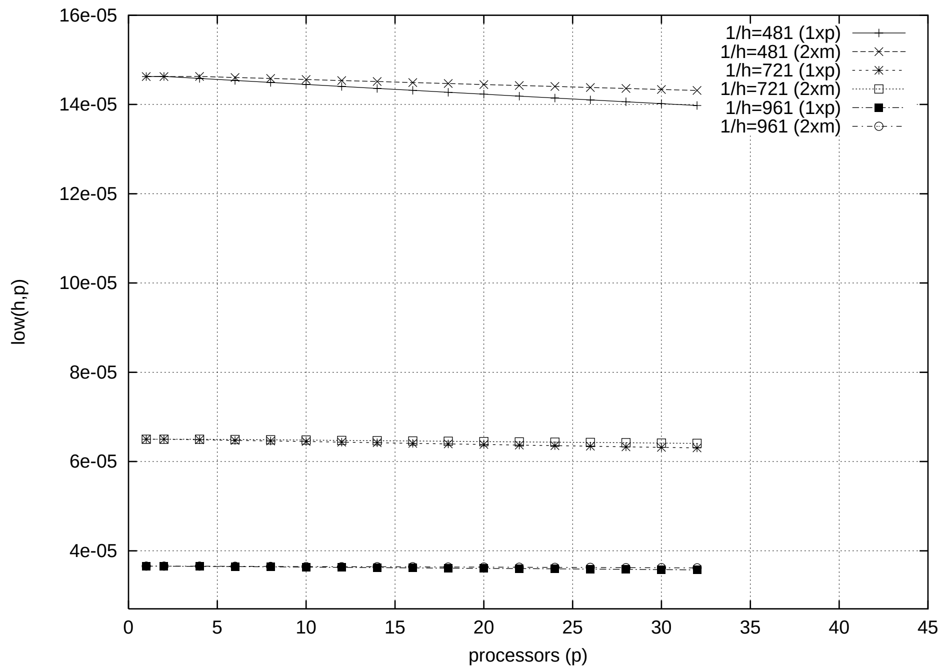

Figure 6. Evolution of the eigenvalue lower bound estimates (5.15) and (5.21) (low(h,p)) as a function of the number of subdomains $p$ ($p \leq 32$), for ParIC(0); $1/h = 481, 721, 961$. $m = p/2$.

of subdomains increases, the decrease of the lower eigenvalue estimate $low(h,p)$, and the increase of the spectral condition number estimate $k(h,p)$ are remarkably slow. The curves for $low(h,p)$ (Fig. 6) display that, for very large problems, $2 \times m$-partitionings would be better than partitioning by stripes. This seems to be contradicted by the evolution of $k(h,p)$ in Fig. 7. This, as well as the jumps of $k(h,p)$ between $p = 1$, $p = 2$, and $p = 4$, comes from the largest eigenvalue estimates (5.11) and (5.17). These estimates are somewhat pessimistic for $p \geq 2$ (see Table I in Section 6). It is a shortcoming of the theory (Lemma 5.1). One makes use of a Gerschgorin disk Theorem (see, e.g., [40]) type argument in order to estimate the minimal value of a parameter $\mu$ such that the matrix $(2 - 1/\mu)P - \mathrm{diag}(A)$ is positive definite [22, Theorem 4.3]. Observe also that, the $\gamma_{i,i}$, defined by (5.2), are inversely proportional to $p_{i,i}$. Now, for $p \geq 2$, and for class 2 gridpoints (class 2 and class 4 gridpoints in the case of $2 \times m$-partitionings), the values of $p_{i,i}$ are smaller than the corresponding values for $p = 1$. In contrast to the case of $p = 1$, the above-mentioned gridpoints are connected to more than two gridpoints with smaller number, which together with the rules given in Fig. 2, explains the decrease of $p_{i,i}$, and hence also the increase of $k(h,p)$.

It is worthwhile to stress that, for $2 \times m$-partitionings, where class 4 gridpoints are connected with 4 gridpoints, the actual largest eigenvalues also (slightly) increases when the (total) number of subdomains extends 4 (see Table I in Section 6). For $p = 2, 4$, the involved orderings are equivalent to the lexicographical one [14], whereas for $p > 4$, *incompatible nodes* appear. For the partitioning by stripes, only the $1 \times 2$-ordering is equivalent to the lexicographical one. The appearance of *incompatible nodes* seems to mostly influence the smallest eigenvalues.

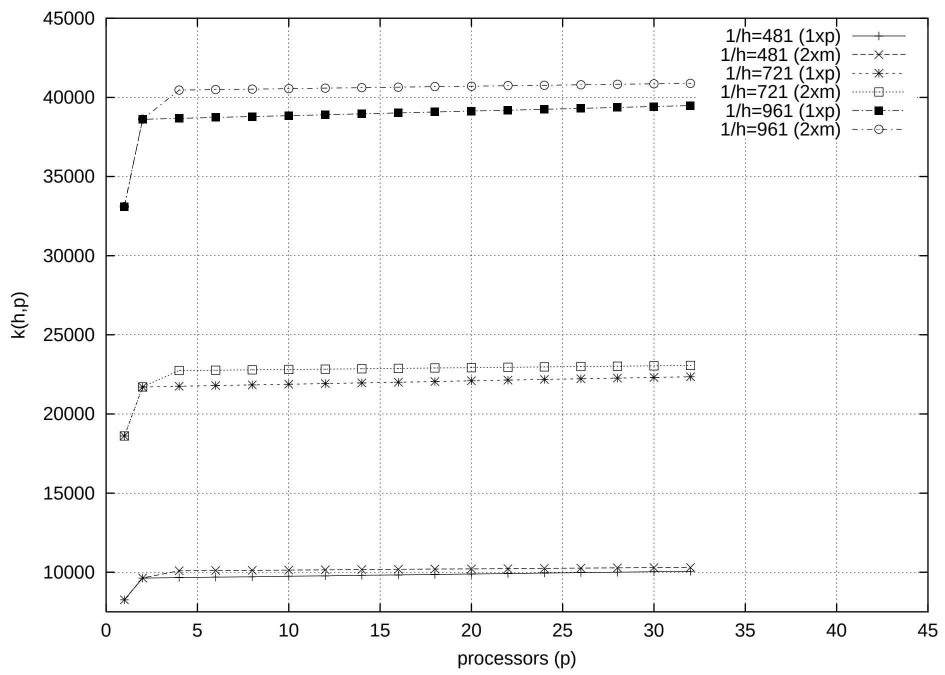

Figure 7. Evolution of the spectral bound estimates (5.16) and (5.22) (k(h,p)) as a function of the number of subdomains $p$ ($p \leq 32$), for ParIC(0); $1/h = 481, 721, 961$. $m = p/2$.

More general PDEs may be handled by adapting our argument, as in [22, Section 5] or [25, Section 5] for perturbed MIC type preconditioners. Upper spectral bound, which extends [22, Theorem 4.3] to higher fill levels, has been obtained in [21, Theorem 3.5]. This bound only involves the entries of the original system matrix, and the ones computed during the incomplete factorization process. It should be noticed that $(\mathbf{e}, R\mathbf{e})$ would decrease as the fill level increases. Lemma 2 then suggests that the smallest eigenvalues of $B^{-1}A$ will very likely increase.

## 6. Numerical Results

The computations have been carried out, in double precision Fortran, on a 16-processor SGI Origin 2000 (195 MHz), using the MPI library for interprocessor (nonblocking) communications. The PCG was executed with the zero vector as initial approximation, and the relative residual error $\|r^{(i)}\|_2 \,/\, \|r^{(0)}\|_2 \leq 10^{-6}$ as convergence criterion. To avoid computing the residual norm $\|r^{(i)}\|_2$ at each iteration, the test was performed only when the preconditioned residual $\gamma_i$ (see Fig. 1) satisfies $\sqrt{\frac{\gamma_i}{\gamma_0}} \leq 10^{-6}$. We have considered the $1 \times p$ (stripes), and the $2 \times m$-partitionings ($m = p/2$). The preconditionings include: ParIC($\ell$) and ParIC*($\ell$).

**Problem 1.** $\alpha = \beta = 1$, , $= \Omega$, *$u(x,y) = x(x-1)y(y-1)e^{xy}$, and $h = 1/(n_y + 1)$.*

To assess the quality of the bounds obtained in Section 5 we give in Table I the extremal eigenvalues, as computed by the PCG process, their theoretical estimates, and the spectral condition number, for ParIC(0). We have used $h^{-1} = 481$, so that the global number of unknowns is $n = 230400$. The estimates have been computed by means of (5.11) and (5.13) for the $1 \times p$-partitionings; and by (5.17) and (5.19) for the $2 \times m$-partitionings. We have also included the number of PCG iterations, the overall computation (wall-clock) time, as well as the parallel speed-up, which is defined as the ratio between the execution time of the parallel algorithm on one processor and the time on $p$ processors.

Table I. Problem 1. $h^{-1} = 481$; $n = 230400$. Extremal eigenvalues ($\lambda_{\min}$ and $\lambda_{\max}$), and their estimates ($\lambda_{\min}^{(e)}$ and $\lambda_{\max}^{(e)}$); spectral condition number ($\kappa$) for the preconditioned matrix $B^{-1}A$ associated with ParIC(0); number of pcg iterations (iter), overall elapsed time (time) in seconds. "Part" denotes the partitioning.

| Part | $\lambda_{\min}$ | $\lambda_{\min}^{(e)}$ | $\lambda_{\max}$ | $\lambda_{\max}^{(e)}$ | $\kappa$ | iter | time | speed-up |
|---|---|---|---|---|---|---|---|---|
| 1 | 1456E-7 | 1463E-7 | 1.207 | 1.207 | 8289 | 372 | 128.72 | 1.00 |
| $1 \times 2$ | 1456E-7 | 1463E-7 | 1.207 | 1.417 | 8289 | 372 | 64.53 | 1.99 |
| $1 \times 4$ | 1452E-7 | 1458E-7 | 1.207 | 1.417 | 8315 | 408 | 28.79 | 4.47 |
| $1 \times 8$ | 1443E-7 | 1449E-7 | 1.207 | 1.417 | 8364 | 409 | 13.04 | 9.87 |
| $1 \times 16$ | 1426E-7 | 1432E-7 | 1.207 | 1.417 | 8465 | 411 | 7.38 | 17.44 |
| $2 \times 1$ | 1456E-7 | 1463E-7 | 1.207 | 1.417 | 8289 | 372 | 64.53 | 1.99 |
| $2 \times 2$ | 1456E-7 | 1463E-7 | 1.207 | 1.478 | 8289 | 374 | 27.46 | 4.69 |
| $2 \times 4$ | 1452E-7 | 1458E-7 | 1.217 | 1.478 | 8382 | 409 | 12.71 | 10.12 |
| $2 \times 8$ | 1443E-7 | 1449E-7 | 1.217 | 1.478 | 8433 | 411 | 7.15 | 18.00 |

All our theoretical bounds, in particular the estimates for the smallest eigenvalue, are in general accurate. They display that, for constant (or smoothly variable) PDE coefficients, the rate of convergence weakly depends on the number of subdomains $p$. This explains the high speed-up observed. This is also borne out from numerical experiments performed in [28, 29], even for some PDEs with strong jumps in the coefficients. Observe that the upper spectral bounds (5.11) and (5.17), which are independent of both the mesh size parameter $h$ and the number of subdomains $p$, reflect the behavior of the actual largest eigenvalues. From the comparison of (5.14) and (5.15) with (5.20) and (5.21), one should expect that $2 \times m$-partitionings give better performance than stripe-partitionings. This is in general the case, at least in terms of number of iterations; see, in particular, Example 2 in [29]. The overall communication time for $2 \times m$-partitionings may be higher than for $1 \times p$-partitionings, because in the former case, the subdomains have more neighbors. Further numerical evidence may be found in [29, Section 4].

It should be emphasized that, for very difficult problems, the convergence rate of level zero incomplete factorization type preconditionings would degrade as the number of subdomains increases (see, e.g., [11, 18, 24, 34]). To resolve this inconvenience, it is mandatory to accept sufficient many fill entries induced by the parallelization renumbering strategy [28, 29].

To compare ParIC($\ell$) and ParIC*($\ell$), we take $\ell = 4, 8$. We also consider the following more difficult boundary value problem.

**Problem 2.** $\Gamma = \{(x, y);\, 0 \le x \le 1,\, y = 0\}$, $h = 1/n_y$,

$$\alpha = \beta = \begin{cases} 100 & \textit{in } (1/4, 3/4) \times (1/4, 3/4) \ , \\ 1 & \textit{elsewhere} \ , \end{cases}$$

$$f(x, y) = \begin{cases} 100 & \textit{in } (1/4, 3/4) \times (1/4, 3/4) \ , \\ 0 & \textit{elsewhere} \ . \end{cases}$$

In Tables II and III, we give the extremal eigenvalues, the spectral condition number, the PCG iteration counts, the overall computation time, and the parallel speed-up. For Problem 2, where $\lambda_{\min}(B^{-1}A)$ is strongly isolated from the rest of the spectrum, we include the second smallest eigenvalue $\lambda_2 = \lambda_2(B^{-1}A)$ as well, and the effective spectral condition number

$$\kappa^{(2)} = \kappa^{(2)}(B^{-1}A) = \frac{\lambda_{\max}(B^{-1}A)}{\lambda_2(B^{-1}A)} \ , \tag{6.1}$$

which governs the superlinear convergence of the PCG method (see e.g. [33, 36, 37]).

We make the following observations.

1. As in the case of level zero incomplete factorization preconditionings (see Table I), the performance depends weakly on the number of subdomains. This is remarkable, because it is achieved without the addition of any coarse-grid correction [35].
2. A high fill level (here $\ell = 8$) is not necessary to achieve better performance, for ParIC and ParIC*. However, note that the linear system is solved only once, so that the gain in the iteration counts is offset by the increase of the incomplete factorization costs. Very ill-conditioned systems may require higher fill level, and/or more elaborated partitionings, for instance, $4 \times 4$-partitionings instead of $2 \times 8$-partitionings.
3. The speed-up observed is high for both preconditioners. ParIC* is not significantly more efficient than ParIC, as one would expect from the fact that ParIC* keeps more low level fill entries. The cancellation of some fill entries does not dramatically degrade the convergence. The rather small gain in the number of iterations, observed with ParIC*, is amortized by the increase of the synchronization phases.

## 7. Conclusions

We have investigated two general approaches for the parallelization of incomplete factorization type preconditionings. The gridpoints are collected into classes, or classes and subclasses, according to the number of subdomains to which they belong. The computation and communication then proceed class by class, and subclass by subclass, if any. Analytical spectral bounds, which agree very well with the actual eigenvalues of the preconditioned systems, have been derived. They show that, for PDEs with constant (or smoothly variable) coefficients, the convergence rate only weakly depends on the number of subdomains. This also holds for some PDEs with strong discontinuities in the variation of the coefficients; and, by ample

Table II. Problem 1. $h^{-1} = 481$; $n = 230400$. Extremal eigenvalues ($\lambda_{\min}$ and $\lambda_{\max}$); spectral condition number ($\kappa$); number of pcg iterations (iter), overall elapsed time (time) in seconds. "Precond" and "Part" denote the preconditioning and the partitioning, respectively.

| Precond | Part | $\lambda_{\min}$ | $\lambda_{\max}$ | $\kappa$ | iter | time | speed-up |
|---|---|---|---|---|---|---|---|
| ParIC(4) | 1 | 1828E-6 | 1.146 | 627 | 115 | 62.69 | 1.00 |
| | $2 \times 1$ | 1787E-6 | 1.177 | 659 | 119 | 34.46 | 1.82 |
| | $2 \times 2$ | 1740E-6 | 1.391 | 799 | 125 | 16.56 | 3.79 |
| | $2 \times 4$ | 1720E-6 | 1.448 | 841 | 127 | 7.07 | 8.87 |
| | $2 \times 8$ | 1630E-6 | 1.448 | 888 | 134 | 4.33 | 14.48 |
| ParIC*(4) | $2 \times 2$ | 1785E-6 | 1.184 | 663 | 119 | 15.88 | 3.95 |
| | $2 \times 4$ | 1734E-6 | 1.190 | 686 | 123 | 7.03 | 8.92 |
| | $2 \times 8$ | 1643E-6 | 1.190 | 725 | 126 | 4.12 | 15.22 |
| ParIC(8) | 1 | 6791E-6 | 1.145 | 168 | 62 | 61.38 | 1.00 |
| | $2 \times 1$ | 6357E-6 | 1.214 | 191 | 66 | 35.25 | 1.74 |
| | $2 \times 2$ | 5459E-6 | 1.425 | 261 | 69 | 18.26 | 3.36 |
| | $2 \times 4$ | 5683E-6 | 1.549 | 273 | 77 | 9.39 | 6.54 |
| | $2 \times 8$ | 5106E-6 | 1.549 | 303 | 81 | 5.17 | 11.87 |
| ParIC*(8) | $2 \times 2$ | 6351E-6 | 1.218 | 192 | 67 | 18.06 | 3.40 |
| | $2 \times 4$ | 5946E-6 | 1.254 | 211 | 69 | 8.69 | 7.06 |
| | $2 \times 8$ | 5294E-6 | 1.254 | 237 | 74 | 4.68 | 13.12 |

experimental evidence, for any PDE, provided that enough fill-in entries, induced by the parallelization strategy, are accepted [28, 29]. It would be interesting to choose the fill level proportional to the number of subdomains, for a fixed mesh size problem. This requires further investigation. Another point which deserves to be explored is the usage of other variants of the basic IC (see, e.g., [1, 2, 4, 5, 20, 25]).

## ACKNOWLEDGEMENTS

We wish to thank Owe Axelsson and the referees for their comments and suggestions to improve the readability of the paper. Thanks are also due to one referee for bringing reference [7] to our attention.

Table III. Problem 2. $h^{-1} = 480$; $n = 230880$. Extremal eigenvalues ($\lambda_{\min}$, $\lambda_2$, and $\lambda_{\max}$); effective spectral condition number ($\kappa^{(2)}$); number of pcg iterations (iter), overall elapsed time (time) in seconds. "Precond" and "Part" denote the preconditioning and the partitioning, respectively.

| Precond | Part | $\lambda_{\min}$ | $\lambda_2$ | $\lambda_{\max}$ | $\kappa^{(2)}$ | iter | time | speed-up |
|---|---|---|---|---|---|---|---|---|
| ParIC(4) | 1 | 1254E-8 | 2291E-6 | 1.167 | 509 | 174 | 90.26 | 1.00 |
| | $2 \times 1$ | 1226E-8 | 2279E-6 | 1.220 | 535 | 146 | 41.53 | 2.17 |
| | $2 \times 2$ | 1194E-8 | 2261E-6 | 1.391 | 615 | 148 | 19.35 | 4.66 |
| | $2 \times 4$ | 1159E-8 | 2217E-6 | 1.448 | 653 | 163 | 8.83 | 10.22 |
| | $2 \times 8$ | 1097E-8 | 2098E-6 | 1.448 | 690 | 170 | 5.22 | 17.29 |
| ParIC*(4) | $2 \times 2$ | 1223E-8 | 2272E-6 | 1.221 | 537 | 146 | 19.11 | 4.72 |
| | $2 \times 4$ | 1163E-8 | 2226E-6 | 1.350 | 606 | 160 | 8.64 | 10.45 |
| | $2 \times 8$ | 1104E-8 | 2106E-6 | 1.350 | 641 | 166 | 4.84 | 18.65 |
| ParIC(8) | 1 | 4721E-8 | 8612E-6 | 1.168 | 136 | 94 | 84.79 | 1.00 |
| | $2 \times 1$ | 4425E-8 | 8482E-6 | 1.258 | 148 | 79 | 40.14 | 2.11 |
| | $2 \times 2$ | 3843E-8 | 8060E-6 | 1.425 | 177 | 87 | 21.51 | 3.94 |
| | $2 \times 4$ | 3861E-8 | 7746E-6 | 1.549 | 200 | 96 | 10.83 | 7.83 |
| | $2 \times 8$ | 3413E-8 | 6903E-6 | 1.549 | 224 | 102 | 5.75 | 14.75 |
| ParIC*(8) | $2 \times 2$ | 4408E-8 | 8407E-6 | 1.259 | 150 | 80 | 20.27 | 4.18 |
| | $2 \times 4$ | 3928E-8 | 7977E-6 | 1.423 | 178 | 89 | 10.24 | 8.28 |
| | $2 \times 8$ | 3516E-8 | 7072E-6 | 1.423 | 201 | 94 | 5.65 | 15.01 |

## REFERENCES

1. Axelsson O. *Iterative Solution Methods*. Cambridge University Press: Cambridge, 1994.
2. Axelsson O, Lindskog G. On the eigenvalue distribution of a class of preconditioning methods. *Numerische Mathematik* 1986; **48**:479–498.
3. Barret RF, Berry M, Chan TF, Demmel J, Donato J, Dongarra JJ, Eijkhout V, Pozo R, Romine C, van der Vorst HA. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM: Philadelphia, 1994.
4. Beauwens R. Modified incomplete factorization strategies. In *Preconditioned Conjugate Gradient Methods,*, Axelsson O, Kolotilina L, (eds). Lectures Notes in Mathematics **1457**. Springer-Verlag: Berlin, 1990; 1–16.
5. Beauwens R, Dujacquier L, Hitimana S, Magolu monga Made M. MILU factorizations for 2-processor orderings. In *Advances in Numerical Methods and Applications $\mathcal{O}(h^3)$,* Dimov IT, Sendov BL, Vassilevski PS, (eds). World Scientific: Singapore, 1994; 26–34.
6. Berman A, Plemmons RJ. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press: New York, 1979.
7. Bridson R, Tang WP. A Structural Diagnosis of Some IC Orderings. *SIAM Journal on Scientific Computing* 2000; **22**:1527–1532.
8. Chan TF, Elman HC. Fourier analysis of iterative methods for elliptic problems. *SIAM Review* 1989; **31**:20–49.
9. Doi S. On parallelism and convergence of incomplete LU factorizations. *Applied Numerical Mathematics* 1991; **7**: 417-436.
10. Doi S, Lichnewsky A. A graph-theory approach for analyzing the effects of ordering on ILU preconditioning. INRIA Report 1452, INRIA-Rocquencourt: France, 1991.
11. Doi S, Washio T. Ordering strategies and related techniques to overcome the trade-off between parallelism and convergence in incomplete factorizations. *Parallel Computing* 1999; **25**:1995–2014.
12. Dongarra JJ, Duff IS, Sorensen DC, van der Vorst HA. *Numerical Linear Algebra for High-Performance Computers*. SIAM: Philadelphia, 1998.
13. van Driessche R. Algorithms for static and dynamic load balancing on parallel computers. Ph.D. thesis, Dept of Computer Science, Katholieke Universiteit Leuven: Belgium, 1995.
14. Duff IS, Meurant GA. The effect of ordering on preconditioned conjugate gradients. *BIT* 1989; **29**:635–657.
15. Duff IS, van der Vorst HA. Developments and Trends in the Parallel Solution of Linear Systems. *Parallel Computing* 1999; **25**:1931–1970.
16. Elman HC. A stability analysis of incomplete LU factorizations. *Mathematics of Computation* 1986; **47**:191–217.
17. Golub GH, van Loan CF. *Matrix Computations* (3rd edn). The John Hopkins University Press: Baltimore, Maryland, 1996.
18. Haase G. Parallel incomplete Cholesky preconditioners based on the nonoverlapping data distribution. *Parallel Computing* 1998; **24**:1685–1703.
19. Hysom D, Pothen A. A scalable parallel algorithm for incomplete factor preconditioning. *SIAM Journal on Scientific Computing*. To appear.
20. Magolu monga Made M. Modified block-approximate factorization strategies. *Numerische Mathematik* 1992; **61**:91–110.
21. Magolu monga Made M. Sparse Approximate Block Factorizations for Solving Symmetric Positive (semi)definite Linear Systems. Ph.D. thesis, Service de Métrologie Nucléaire, Université Libre de Bruxelles: Belgium, 1992.
22. Magolu monga Made M. Analytical bounds for block approximate factorization methods. *Linear Algebra and its Applications* 1993; **179**:33–57.
23. Magolu monga Made M. Ordering strategies for modified block incomplete factorizations. *SIAM Journal on Scientific Computing* 1995; **16**:378–399.
24. Magolu monga Made M. Implementation of parallel block preconditionings on a transputer-based multiprocessor. *Future Generation Computer Systems* 1995; **11**:167–173.
25. Magolu monga Made M. Taking advantage of the potentialities of dynamically modified block incomplete factorizations. *SIAM Journal on Scientific Computing* 1998: **19**:1083–1108.
26. Magolu monga Made M, Polman B. Efficient planewise like preconditioners to cope with 3D problems. *Numerical Linear Algebra with Applications* 1999; **6**:379–406.
27. Magolu monga Made M, Valkenberg R, Beauwens R, Warzée G. A synthesis of available automatic mesh partitioning techniques. DOMINOS Report D1.1.2. Service des Milieux Continus, Université Libre de Bruxelles, April 1998.
28. Magolu monga Made M, van der Vorst HA. Parallel incomplete factorizations with pseudo-overlapped subdomains. *Parallel Computing*. To appear.
29. Magolu monga Made M, van der Vorst HA. ParIC : A Family of Parallel Incomplete Cholesky

Preconditioners. In *High Performance Computing and Networking,* Bubak M, Afsarmanesh H, Williams R, Hertzberger B (eds). Proceedings of the HPCN Europe 2000 Conference, Amsterdam. Lecture Notes in Computer Science, **1823**. Springer-Verlag: Berlin, 2000; 89–98.

30. Meijerink JA, van der Vorst HA. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation* 1977; **31**:148–162.
31. Meijerink JA, van der Vorst HA. Guidelines for the usage of incomplete decompositios in solving sets of linear equations as they occur in practical problems. *Journal of Computational Physics* 1981; **44**:134–155.
32. Nakamura S. *Computational Methods in Engineering and Science*. John Wiley & Sons: New York, 1977.
33. Notay Y. On the convergence rate of the conjugate gradients in the presence of rounding errors. *Numerische Mathematik* 1993; **65**:301–317.
34. Notay Y. An efficient Parallel Discrete PDE Solver. *Parallel Computing* 1995; **21**:1725–1748, 1995.
35. Notay Y, Van de Velde A. Coarse grid acceleration of parallel incomplete preconditioners. In *Iterative Methods in Linear Algebra II,* Margenov S, Vassilevski P (eds). IMACS series in Computational and Applied Mathematics 1996; **3**:106–130.
36. van der Sluis A. The convergence behaviour of conjugate gradients and ritz values in various circumstances. In *Iterative Methods in Linear Algebra,* Beauwens R, de Groen P (eds). North-Holland: Amsterdam, 1992; 49–66.
37. van der Sluis A, van der Vorst HA. The rate of convergence of conjugate gradients. *Numerische Mathematik* 1986; **48**:543–560.
38. van der Sluis A, van der Vorst HA. The convergence behavior of Ritz values in the presence of close eigenvalues. *Linear Algebra and its Applications* 1987; **88/89**:651–694.
39. Smith BF, Bjørstad PE, Gropp D. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press: Cambridge, 1996.
40. Varga RS. *Matrix Iterative Analysis*. Prentice Hall: Englewood Cliffs, 1962.
41. van der Vorst HA. Iterative solution methods for certain sparse linear systems with a non-symmetric matrix arising from PDE-problems. *Journal of Computational Physics* 1981; **44**:1–19.
42. van der Vorst HA. Preconditioning by incomplete decompositions. PhD thesis, Utrecht University, the Netherlands, 1982.
43. van der Vorst HA. A vectorizable variant of some ICCG methods. *SIAM Journal on Scientific and Statistical Computing* 1982; **3**:350–356.
44. van der Vorst HA. Large tridiagonal and block tridiagonal linear systems on vector and parallel computers. *Parallel Computing* 1987; **5**:54–54.
45. van der Vorst HA. High performance preconditioning. *SIAM Journal on Scientific and Statistical Computing* 1989; **10**:1174–1185.
46. van der Vorst HA, Sleijpen GLG. The effect of incomplete decomposition preconditioning on the convergence of Conjugate Gradients. In *Incomplete Decomposition (ILU)–Algorithms, Theory and Applications,* Hackbusch W, Wittum W (eds). Notes on Numerical Fluid Mechanics **41**. Vieweg: Braunschweig, 1993; 179–187.