Parallel Implementation of an Explicit Formulation of the Multiplicative Schwarz Preconditionner

Guy Antoine Atenekeng Kahou
Department of Computer Science
University of Yaounde 1
PO Box 812 YAOUNDE, CAMEROON
E-mail: gaatenek@irisa.fr

Emmanuel Kamgnia
Department of Computer Science
University of Yaounde 1
PO Box 812 YAOUNDE, CAMEROON
E-mail: erkamgnia@yahoo.fr

Bernard Philippe
INRIA/IRISA
Campus de Beaulieu
35042 RENNES Cedex, FRANCE
E-mail: philippe@irisa.fr

Abstract - We present an explicit formulation of the splitting associated with the Mutiplicative Schwartz iteration. We give an implementation of the new formulation on a parallel computer and show its advantages when used as a preconditionner for a Krylov method. In the paper the implementation is described in PETCs standard.

Keywords— **Domain decomposition, Multiplicative Schwarz preconditionner, iterative methods, PETSc library, parallel computing.**

I. Introduction

Domain decomposition provides a class of divide-and-conquer methods suitable for the solution of linear or non-linear systems of equations arising from the discretization of partial differential equations. For linear systems, domain decomposition methods can be viewed as preconditioners for Krylov subspace techniques.
In preconditioning methods, which is our interest in this article, domain decomposition refers to the process of subdividing the solution of a large linear system into smaller problems whose solutions can be used to produce a preconditioner (or solver) for the system of equations that results from discretizing the PDE on the entire domain or more generally from any sparse matrix. In our work, we consider the latter and we suppose that the domain decomposition is with overlapping.
Traditionally, there are two classes of iterative methods that derive from domain decomposition with overlapping : Additive Schwarz and Multiplicative Schwarz. When using these two methods as solvers, the convergence rates are often slow and convergence is mainly guarantied for symmetric positive definite matrices and M-matrices [BEN 99]. For that reason, the particular interest of Schwarz methods is as preconditioner of Krylov subspace methods since they can be efficient even when they would not converge as a full method.
When used as preconditioners, one is interested in deriving an explicit and useful expression of the preconditioner. For the Additive Schwarz method such an expression exists. To our knowledge, no explicit expression of the preconditionner for the Multiplicative Schwarz method exists. In a previous work we derived such an explicit formulation. In this paper, we discuss the implementation of the new formulation on a parallel computer.

II. Domain decomposition of a sparse matrix and notations

Let us consider a sparse matrix $A \in R^{n\times n}$. The pattern of A is the set $P = \{(k,l)|a_{k,l} \neq 0\}$ which is the set of the edges of the graph $G = (W,P)$ where $W = \{1,...,n\} = [1:n]$ is the set of vertices.
Definition II.1: A domain decomposition of matrix A into p subdomains is defined by a collection of sets of integers $W_i \subset W = [1:n]$, $i = 1,...,p$ such that :

$$\left\{ \begin{array}{l} |i-j| > 1 \Longrightarrow W_i \cap W_j = \emptyset \\ P \subset \bigcup_{i=1}^{p} (W_i \times W_i) \end{array} \right.$$

Following this definition, a domain decomposition can be considered as resulting from a graph partitioner but with potential overlap between domains. For the rest of our discussion, we shall suppose that a graph partitioner has been applied and has resulted in p sets W_i whose union is W,

$W = [1 : n]$.
We shall denote by $L_i = \oplus_{j \in W_i}(e_j)$ the vector space of R^n of all the vectors with zero components for every index $j \notin W_i$. Let m_i be the dimension of L_i. The orthogonal projector onto L_i is defined by the sub-identity matrix I_i of order $n \times n$ whose diagonal elements are set to one if the corresponding node belongs to W_i and to zero otherwise. We define the matrix,

$$A_i^n = I_i A I_i, \tag{1}$$

which is an extension to the whole space, of the restriction of A to L_i. We also define the complement sub-identity matrix $\bar{I}_i = I - I_i$ and the matrix,

$$\bar{A}_i = A_i^n + \bar{I}_i. \tag{2}$$

We assume thereafter that all the matrices $\bar{A}_i$, for $i = 1, \cdots, p$ are non singular. The generalized inverse A_i^+ of A_i^n satisfies $A_i^+ = I_i \bar{A}_i^{-1} = \bar{A}_i^{-1} I_i$.
Proposition II.1: Definition II.1 implies that for any $i, j \in \{1, ..., p\}$, the following is true :

$$|i - j| > 2 \Rightarrow I_i A I_j = 0.$$

Proof: Let $(k, l) \in W_i \times W_j$ such that $a_{k,l} \neq 0$. Since $(k, l) \in P$, there exists $m \in \{1...n\}$ such that $k \in W_m$ and $l \in W_m$; therefore $W_i \cap W_m \neq \emptyset$ and $W_j \cap W_m \neq \emptyset$. Consequently, from Definition II.1, $|i - m| \leq 1$ and $|j - m| \leq 1$, which implies $|i - j| \leq 2$. ■
Let us introduce a special situation which is often satisfied and which wil bring some simplification in the sequel.
Definition II.2: The domain decomposition is with *weak overlap* if and only if, for any $i,\ j \in \{1, ..., p\}$ the following is true

$$|i - j| > 1 \Rightarrow I_i A I_j = 0.$$

The set of unknowns which represents the overlap is defined by the set of integers $J_i = W_i \cap W_{i+1}, i = 1, ..., p-1$, and size this overlap by s_i . Similarly to (1) and (2), we define

$$C_i^n = O_i A O_i, \tag{3}$$

and

$$\bar{C}_i = C_i^n + \bar{O}_i, \tag{4}$$

where $O_i \in R^{n \times n}$ is sub-identity matrix whose diagonal elements are set to one if the corresponding node belongs to J_i and to zero otherwise, and $\bar{O}_i = I - O_i$.
Example II.1: Figure 1 displays an example of a domain decomposition for a matrix in the case where all W_i are intervals of integers.
There is a close connection between a block tridiagonal structure and a domain decomposition. For instance, in this example, if, in order to transform A into a block tridiagonal matrix, we assume that all the blocks $A_{i,i+2}$ and $A_{i+2,i}$ are zeros, the domain decomposition is obtained by defining

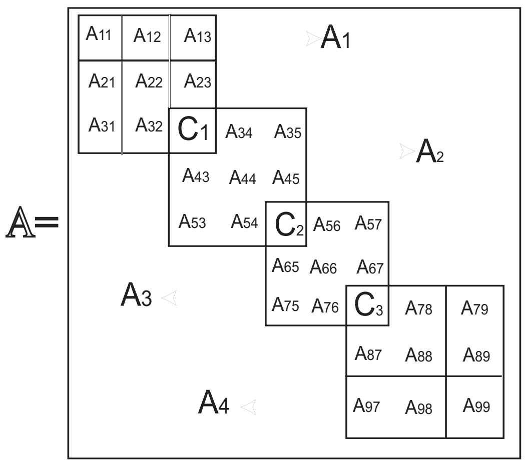

Fig. 1. A matrix domain decomposition with block overlaps

the domains with three consecutive blocks and the overlaps correspond to only one block. One can easily verify that such an overlap is a weak overlap. If the domains were defined with only two blocks and the domains overlap on one block, the overlap would not be a weak overlap.

In the definitions, the set of integers defining the subdomains are not assumed to be intervals although this is often the case as in the example. However, when considering other cases like a red-black ordering of blocks, it is important to include the general case. Nevertheless, it is always possible to recover a special case by renumbering the unknowns.

III. Multiplicative Schwarz : Classical formulation

The goal of Multiplicative Schwarz methods is to iteratively solve a linear system

$$Ax = b \tag{5}$$

where matrix A is decomposed into overlapping subdomains as described in the previous section. The iteration consists of solving the original equation in sequence on each subdomain. This is a well-known method ; for more details, see for instance [BEN 99], [HAC 99], [MEU 99], [SAA 03], [SMI 96]). In this section, we present the main properties of the iteration and exhibit an explicit formulation of the corresponding matrix splitting.

Let x^k be the current iterate and $r^k = b - Ax^k$ the corresponding residual; the method proceeds as follows.

It builds p sub-iterates $x^{k+i/p}(i = 1, ..., p)$ and their corresponding residuals $r^{k+i/p} = b - Ax^{k+i/p}$ by the following recursion

$$\left\{\begin{array}{l} x^{k+1/p} = x^k + A_1^+ r^k \\ \quad \text{and } r^{k+1/p} = r^k - AA_1^+ r^k, \\ x^{k+2/p} = x^{k+1/p} + A_2^+ r^{k+1/p} \\ \quad \text{and } r^{k+2/p} = r^{k+1/p} - AA_2^+ r^{k+1/p}, \\ \vdots \\ x^{k+1} = x^{k+(p-1)/p} + A_p^+ r^{k+(p-1)/p} \\ \quad \text{and } r^{k+1} = r^{k+(p-1)/p} - AA_p^+ r^{k+(p-1)/p}. \end{array}\right. \tag{6}$$

It follows that :

$$r^{k+1} = (I - AA_p^+)...(I - AA_1^+)r^k. \tag{7}$$

This method corresponds to a relaxation iteration defined by some splitting $A = M - N$ such that the matrices iterating on the residuals and on the errors are respectively

$$\begin{array}{rcl} NM^{-1} & = & (I - AA_p^+)\cdots(I - AA_1^+), \\ \text{and} & & \\ M^{-1}N & = & A^{-1}NM^{-1}A \\ & = & (I - A_p^+A)\cdots(I - A_1^+A). \end{array}$$

The convergence of this iteration is proven for M-matrices and for symmetric positive definite matrices (eg. see [BEN 99]).
Now, let us suppose that the goal is to consider another iterative method but preconditioned by one step of the Multiplicative Schwarz method. For that purpose, it is necessary to define

$$y = M^{-1}Ax \text{ or } y = AM^{-1}x$$

depending on the side of the preconditionning, for any vector x and where M is the matrix characterized by the previous splitting. From the expression of $M^{-1}N$ and NM^{-1} we can derive an expression of $M^{-1}A$ or AM^{-1} as follows :

$$M^{-1}A \quad = \quad I - (I - A_p^+A)\cdots(I - A_1^+A), \tag{8}$$

or

$$AM^{-1} \quad = \quad I - (I - AA_p^+)\cdots(I - AA_1^+). \tag{9}$$

IV. Multiplicative Schwarz preconditionner

A. Explicit formulation

Theorem IV.1: Let $A \in R^{n\times n}$ be decomposed into p subdomains as described in section II such that all the matrices $\bar{A}_i$, and the matrix C_i for $i = 1, \cdots, p$ are non singular. The Multiplicative Schwarz preconditionner matrix M^{-1} can be explicitly expressed by :

$$M^{-1} = \bar{A}_p^{-1}\bar{C}_{p-1}\bar{A}_{p-1}^{-1}\bar{C}_{p-2}\cdots\bar{A}_2^{-1}\bar{C}_1\bar{A}_1^{-1} \tag{10}$$

where for $i = 1, \cdots, p$ matrix $\bar{A}_i$ correspond to the block A_i and for $i = 1, \cdots, p-1$ matrix $\bar{C}_i$ to the overlap C_i when completed by identity in both cases (see the notations introduced in section II).

Proof: See [ATE 05]. ■

Let us define the following partition of matrix A_i

$$A_i = \begin{pmatrix} B_i & F_i \\ E_i & C_i \end{pmatrix},$$

which allows us to characterize the matrix $N = M - A$.

Proposition IV.1: The matrix N defined by the multiplicative Schwarz splitting $A = M - N$ can be expressed as follows:

$$\left\{\begin{array}{ll} N_{ij} & = G_i \cdots G_{j-1}B_j \; if \; j > i+1 \\ N_{ii+1} & = G_iB_{i+1} - [F_i \; 0] \\ N_{ij} & = 0 \; otherwise \end{array}\right. \tag{11}$$

where $G_i = (F_iC_i^{-1} \; 0)$ for i , $j = 1, ..., p-1$.

When the domain decomposition is with weak overlap, expression (11) becomes:

$$\left\{\begin{array}{ll} N_{ii+1} & = G_iB_{i+1} - [F_i \; 0] \\ N_{ij} & = 0 \; otherwise \end{array}\right. \tag{12}$$

where $G_i = (F_iC_i^{-1} \; 0)$ for i , $j = 1, ..., p-1$.

Proof: See [ATE 05]. ■

From, this expression, we can claim that N is rank deficient. Actually it is nilpotent.

Corollary IV.1: In the splitting $A = M - N$ associated with the Multiplicative Schwarz method, matrix N is of rank $r \le \sum_{i=1}^{p-1} s_i$.

Proof: The proof is obvious when the decomposition is with weak overlap. For general case, see [ATE 05]. ■

That property implies that, in exact arithmetic, a Krylov method, preconditioned by the Multiplicative Schwarz iteration, computes the exact solution at the $(r+1)^{th}$ iteration, at the latest [ATE 05].

B. Symmetrization of M

Even when matrix A is symmetric, the preconditioned conjugate gradient method cannot be used directly since the Multiplicative Schwarz preconditioner is not symmetric. However, it can easily be symmetrized by including a second sweep corresponding to apply M^{-T} to current residual r_{k+1}. It can be written as follows :

$$\left\{\begin{array}{l} x_{k+1} = x_k + M^{-1}r_k \\ x_{k+2} = x_{k+1} + M^{-T}r_{k+1} \end{array}\right.$$

Since $r_{k+1} = NM^{-1}r_k$, we have

$$\begin{aligned} x_{k+2} &= x_k + M^{-1}r_k + M^{-T}NM^{-1}r_k \\ &= x_k + M^{-T}(M^T + N)M^{-1}r_k \\ &= x_k + M^{-T}(M^T + M - A)M^{-1}r_k. \end{aligned}$$

We deduce that the new preconditioning matrix (the one corresponding to two sweeps up and down) is such that:

$$M^{-1} = M^{-T}(M^T + N)M^{-1} = M^{-T}(M^T + M - A)M^{-1}$$

It can be shown [BEN 99] that when A is s.p.d., the preconditionner M is s.p.d. as well.

C. Advantages of the explicit formulation

In the classical expression (6) of the Multiplicative Schwarz iteration the computation of the two steps

$$\begin{aligned} x_{k+1} &= x_k + M^{-1}r_k \text{ and} \\ r_{k+1} &= b - Ax_{k+1}, \end{aligned}$$

is carried out recursively through the domains whereas the explicit formulation decouples the two computations. The computation of the residual is therefore more easily parallelized since it is withdrawn from the recursion. Another advantage of the explicit expression arises when it is used as a preconditioner of a method already coded in a library. In such a case, the user is supposed to provide a code for the procedure $x \to M^{-1}x$. Since the method computes the residual, the classical algorithm computes it once more.
We now show that the number of operations involved in both approaches remains roughly the same, although with a slight advantage to the explicit formulation.
Let us denote by $C(p)_{cla}$ the cost of one iteration using the classical formulation. One can verify that

$$C(p)_{cla} = \sum_{i=1}^{p}(t_i + p_i), \tag{13}$$

where

$$\begin{cases} t_i &= \text{\# of flops for solving in subdomain i :} \\ & x^{k+i/p} = x^{k+(i-1)/p} + A_i^+ r^{k+(i-1)/p}, \\ p_i &= \text{\# flops for multiplying by } A \text{ in subdomain i :} \\ & r^{k+i/p} = r^{k+(i-1)/p} - A(A_i^+ r^{k+(i-1)/p}). \end{cases}$$

Let us also denote by $C(p)_{exp}$ the number of operations for computing $x \to M^{-1}x$, using the explicit form of the Multiplicative Schwarz preconditionner :

$$C(p)_{exp} = \sum_{i=1}^{p-1}(t_i + \tau_i) + t_p, \tag{14}$$

where τ_i is the number of flops for multiplying a vector in subdomain i by C_i. The number of operations $C(A)$ in the computation of a residual, which involves the multiplication by matrix A, satisfies the relation $C(A) = \sum_{i=1}^{p} q_i - \sum_{i=1}^{p-1} \tau_i$ where q_i is the number of operations involved in the multiplication by block A_i. Since $q_i < p_i$ (they are usually close numbers), we obtain that

$$C(p)_{cla} > C(p)_{exp} + C(A). \tag{15}$$

D. Red-Black ordering (M and N)

The Multiplicative Schwarz method clearly is not suitable for parallelism since the recursion between blocks prevents independent calculations. The classical way to overcome the drawback is to relax part of the recursion by a Red-Black coloring. If we remember our assumption that overlap only occurs between consecutive blocks, like in Figure 1, all the blocks of odd numbers can be used in parallel and then the update is performed with all the blocks of even numbers. If we consider a reordering of the unknowns which labels first the components corresponding to the odd domains and then the even ones, it can easily be shown that the new preconditioner is still a Multiplicative Schwarz method but with only two domains ; the method becomes an Alternative Schwarz method. In such a situation the overlap is the union of all the elementary overlaps and therefore the total dimension of the overlap does not change.

V. Parallel Implementation

A. Expressing parallelism

The parallel architecture in mind is a network of p processors $(P_i)_{1,p}$ which communicate through message passing primitives. Let us consider that p blocks are determined in the matrix A so that their order is approximatively the same. Block A_i is stored in the memory of processor P_i. The vectors are stored accordingly ; it is therefore necessary to maintain the consistency between the two copies of the components corresponding to the overlaps.
Matrix A may be considered as a sum of p blocks B_i ($i = 1, \cdots, p$) where $B_p = A_p$ and any other block B_i is the block A_i from which the overlapping block C_i is zeroed as shown in Figure 2.

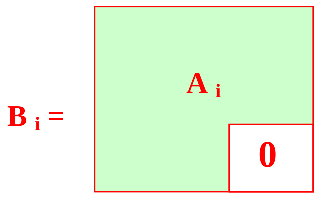

Fig. 2. Elementary block for the matrix multiplication

Therefore the matrix-vector multiplication $x \longrightarrow Ax = \sum_{i=1}^{p} \bar{B}_i x$ can be performed by the following algorithm :

```
Code for Processor $P_i$
  { processor $P_i$ owns $A_i$ and $x_i$ }
    $z_i := B_i x_i$ ;
    if $i < p$,
      send $z_i^3$ to processor $P_{i+1}$ ;
    end ;
    if $i > 1$,
      send $z_i^1$ to processor $P_{i-1}$ ;
    end ;
    if $i < p$,
      receive $f$ from processor $P_{i+1}$ ;
      $z_i^3 := z_i^3 + f$
    end ;
    if $i > 1$,
      receive $g$ from processor $P_{i-1}$ ;
      $z_i^1 := z_i^1 + g$
    end ;
  { processor $P_i$ owns $z_i$ }
```

Parallel application of the operator.

where x_i is the subvector of x corresponding to the i-block limits and where vector z_i is partitioned into $z_i^T = \left({z_i^1}^T, {z_i^2}^T, {z_i^3}^T\right)$ accordingly to the overlap with the neighboring blocks.

The application of the preconditionner, corresponding to one iteration of the Multiplicative Schwarz method, involves a recursion which harms the parallel behaviour. One remedy, as mentioned in Section IV-D is the Red-Black adaptation. Nevertheless, since the blocks are individually stored on the processors, the parallel expression allows to handle large problems. By partitionning the vectors accordingly to the block structure, the procedure on Processor i ($i = 1, \cdots, p$) is expressed by :

```
Code for Processor $P_i$
  { processor $P_i$ owns $A_i$ and $x_i$ }
    if $i > 1$,
      receive $y_{i-1}^3$ from processor $P_{i-1}$ ;
      $x_i^1 := y_{i-1}^3$ ;
    end ;
    Sove $A_i y_i = x_i$ ;
    if $i < p$,
      $y_i^3 := C_i y_i^3$ ;
      send $y_i^3$ to processor $P_{i+1}$ ;
    end ;
    if $i > 1$,
      send $y_i^1$ to processor $P_{i-1}$ ;
    end ;
    if $i < p$,
      receive $y_{i+1}^1$ from processor $P_{i+1}$ ;
      $y_i^3 := y_{i+1}^1$ ;
    end ;
  { processor $P_i$ owns $y_i$ }
```

Preconditioning step : Solve $My = x$.

The use of the preconditionner requires being able to solve linear systems for every matrix A_i. When feasible, the simplest approach consists in LU-factorizing all the matrices. This can be done in a parallel pre-processing step by invoking on each processor a sparse solver such as `SuperLU` [LI 99] or `MUMPS` [AME 00].

In the case of weak overlap, the matrix $N = M - A$ has a very simple shape as indicated in relation (12). In such case, the matrix-vector multiplication $t = Nx$ is easily parallelized. To be more specific, let us partition block A_i and the corresponding vectors x_i and t_i as follows :

$$A_i = \begin{pmatrix} C_i & B_i^{12} & 0 \\ B_i^{21} & B_i^{22} & F_i^2 \\ 0 & E_i^2 & C_{i+1} \end{pmatrix},$$

$$x_i = \begin{pmatrix} x_i^1 \\ x_i^2 \\ x_i^3 \end{pmatrix} \text{ and } t_i = \begin{pmatrix} t_i^1 \\ t_i^2 \\ t_i^3 \end{pmatrix},$$

for $= i = 1, \cdots, p-1$. Notice the zero sub-blocks which appear in that decomposition because of the weak overlap. Block A_p is accordingly defined by omiting the third index since C_p does not exist. In that situation, it is easy to prove that the multiplication of vector x by matrix N is implemented by the following program :

```
Code for Processor $P_i$
  { processor $P_i$ owns $A_i$ and $x_i$ }
    $y_i := B_i^{12} x_i^2$ ;
    if $i > 1$,
      send $y_i$ to processor $P_{i-1}$ ;
    end ;
    if $i < p$,
      receive $y_{i+1}$ from processor $P_{i+1}$ ;
      solve $C_i z = y_{i+1}$ ;
      $t_i^2 := F_i^2 z$ ;
      $t_i^1 := 0$ ;
      $t_i^3 := 0$ ;
    else
      $t_p := 0$ ;
    end ;
  { processor $P_i$ owns $t_i$ }
```

Parallel Multiplication $t = N \times x$ (weak overlap).

Since $M^{-1}A = I - M^{-1}N$, it can be of interest to compare the CPU-times of the multiplication of a vector by matrix A and of the multiplication by matrix N.

B. Test problem

The test matrix is obtained from the discretization of a 3-D flow fracture network [MUS 05]. Each fracture is regarded as a medium 2D and the domain is a collection of such fractures as shown in Figure 3. The mathematical model is

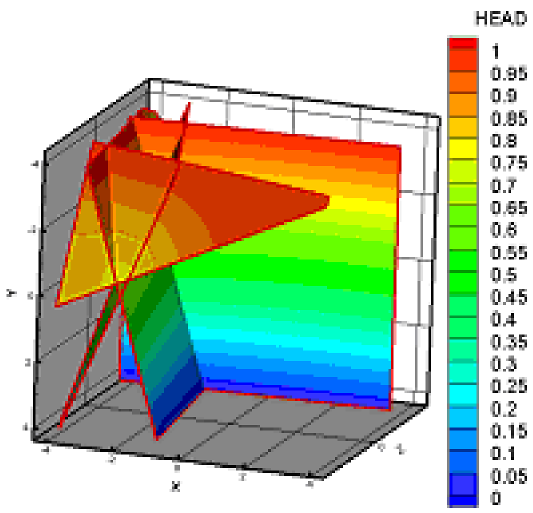

Fig. 3. Example of Network of fractures

the flow expressed with the Darcy's law :

$$\left\{ \begin{array}{rcl} \nabla . q & = & 0, \\ q & = & -T.\nabla h, \end{array} \right. \tag{16}$$

where q and T are respectively the flow and the transitivity integrated upon the thickness of the fracture and h is the load. The system (16) is discretized by the Mixed Hybrid Finite Element method which ends up as a linear system with a sparse symmetric positive definite matrix of order $n = 80,231$ with $n_{nz} = 397,051$ non-zero entries. The pattern of the matrix is displayed in Figure 4. The un-

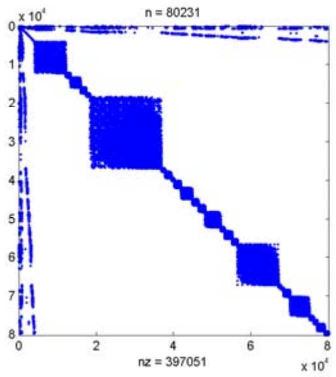

Fig. 4. Pattern of the matrix (original ordering)

knowns are then renumbered with the Symmetric Reverse Cuthill-McKee permutation which reduces the bandwidth of the matrix (half-bandwidth is $b = 671$). We consider three block partitions on the permuted matrix which all satatisfy the property of weak overlap.

The first and the second partition are based on ten domains. The dimension of the overlaps are respectively 700 and 2001. The partitions are defined in the Tables I) and II)

Block #	1	2	3	4	5
First row	1	8000	16000	24000	32000
Last row	10000	18000	26000	34000	42000

Block #	6	7	8	9	10
First row	40000	48000	56000	64000	72000
Last row	50000	58000	66000	74000	80231

TABLE I
BLOCK PARTITION # 1

Block #	1	2	3	4	5
First row	1	8701	17401	26101	34801
Last row	9400	18100	26800	35500	44200

Block #	6	7	8	9	10
First row	43501	52201	60901	69601	78301
Last row	52900	61600	70300	79000	80231

TABLE II
BLOCK PARTITION # 2

The third partition is based on 4 blocks with an overlap equal to 701.

C. Experiments

The experiments were run on a network of nodes connected through a Giga-bit Ethernet switch. Each node is a bi-processor Sun Fire V20z (Opteron at 2.2GHz) with a 2 GigaByte local memory. The tested primitives are :

A) Factor $(A_i)_{1,p}$ *:* This pre-processing step is entirely parallel. On each processor, the step assembles first A_i from the two blocks B_i and C_i (see Figure 2) ; then, the block A_i is factorized by the SuperLU routine.

B) Factor $(C_i)_{1,p-1}$ *:* This pre-processing step is parallel but run on $p-1$ processors. It includes the factorization of the overlapping block C_i by the SuperLU routine and the extraction of block F_i from block A_i.

C) $y := Ax$ *:* This step is parallel.

D) $y := M^{-1}x$ *:* This step is almost sequential (recursion on the blocks).

E) $y := Nx$ *:* This step is parallel but on $p-1$ processors. It must be compared to step C).

F) $y := AM^{-1}x$ *:* This step corresponds to pipelining steps D) and C). Step D) is dominant and therefore the overall is poorly parallelized.

G) $y := x - NM^{-1}x$ *:* This step corresponds to pipelin-

Block #	1	2	3	4
First row	1	19291	38591	57891
Last row	19991	39291	58591	80231

TABLE III
BLOCK PARTITION # 3

Partition #	# 1	# 2	# 3
Number of processors	10	10	4
A) Factor $(A_i)_{1,p}$	4.3157	1.3738	3.3079
B) Factor $(C_i)_{1,p-1}$	0.0193	0.0087	0.0116
C) $y := Ax$	0.0023	0.0016	0.0024
D) $y := M^{-1}x$	0.0676	0.0830	0.0658
E) $y := Nx$	0.0022	0.0017	0.0022
F) $y := AM^{-1}x$	0.0688	0.0545	0.0676
G) $y := x - NM^{-1}x$	0.0693	0.0548	0.0804

TABLE IV
TIMINGS (S)

Partition #	# 1	# 2	# 3
Number of iterations	41	32	23
Time (s)	3.74	2.49	2.64

TABLE V
CONVERGENCE FOR GMRES

ing steps D) and E). Step D) is dominant and therefore the overall is poorly parallelized.

The programs are C-coded and are written under the standard of the library PETSc [BAL 04]. Timings are reported in Table IV.

It can be first noticed that the pre-processing step A) is time consuming and very dependent on the block partitionning. For instance, the block definition of Test # 1 is much less efficient than the block definition of Test # 2. Moreover, the assembly part becomes important for large overlappings.

The timings of Steps C) to G) show that the preconditioning step D) is dominant. This is the effect of the recursion between the blocks which slows down the computation. In the tests, for each block a preliminary reordering by nested dissection was considered.

The last information which can be drawn for the table of timings is that Steps C) and E) are very much equivalent. Therefore Steps F) and G) are also close. These tests are not sufficient to decide if there are situations where the formulation corresponding to Step G) can be more efficient than the direct approach of Step F) when the partition is of weak overlap.

To illustrate the behaviour of the preconditionner, the last test embeds the procedures in the GMRES solver of the PETSc library. The condition number of the solved system is of order 10^5. The maximum size of the basis is chosen as 40 and the tolerance for the precision is fixed as 10^{-12}. When no precondionner is used, the method stagnates. When the Multiplicative Schwarz preconditionner is used, we report in Table V the number of iterations to reach the convergence for the already defined three block partitions. Times are also reported ; they do not include the pre-processing step. In these tests, we replaced the nested dissection reordering on each block by the minimum degree reordering ; this decreases the time spent in each iteration.

It appears that the tests give similar results. The fastest is the test corresponding to the smallest blocks and smallest overlaps since that decreases the time spent in each iteration and while it does not deteriorate too much the convergence. The convergence evolution is displayed in Figure 5.

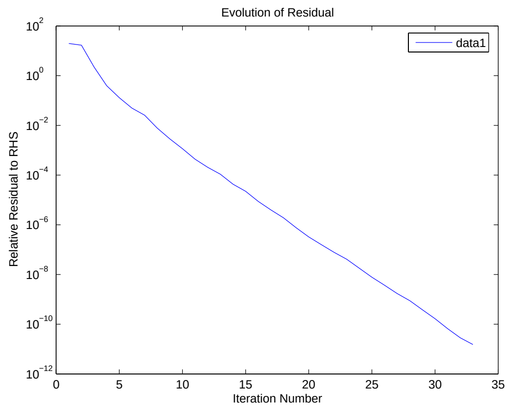

Fig. 5. Convergence of GMRES (Partition # 2)

These tests provide a preliminary study. To obtain a more precise picture, tuning of the codes is mandatory ; tests on bigger matrices must also be considered. Nevertheless, they prove that the explicit expression of the Multiplicative Schwarz allows an easy use of the preconditionner in the framework of the library PETSc.

VI. CONCLUSION

The Multiplicative Schwarz is a very efficient preconditionner especially for Krylov methods.

In this work we have shown how the explicit formulation of the Multiplicative Schwarz preconditionner, previously described in [ATE 05], can be implemented on a parallel computer. By decoupling the application of the preconditioner and the computation of the residual, we obtain an algorithm which is better suited for parallelism and for black-box solvers.

The first experiments exhibited in this paper prove that this approach is of interest. A common situation of block partitionning, which is called of weak overlap, provides an alternative expression of the combination of the two steps of the operator and preconditionner application. A fine tuning of the implementation is necessary to have a better analysis of the advantages of this new formulation. We expect to characterize cases where this formulation is superior.

Although the Additive Schwarz preconditioner is often preferred for its ability to be parallelized, it has a slower convergence rate. We have worked on an alternative expression of the multiplicative version which might change the conclusion.

ACKNOWLEDGEMENTS

This work was supported by the French ministry of Foreign Affairs under the SARIMA program. The computer used for the numerical runs is part of the French GRID5000 project.

REFERENCES

[AME 00] MUMPS, http://www.enseeiht.fr/lima/apo/MUMPS/, March 2000.

[ATE 05] ATENEKENG KAHOU G.-A., KAMGNIA E. , PHILIPPE B., An explicit formulation of the multiplicative Schwarz preconditionner, Research report, INRIA, 2005.

[BAL 04] BALAY S., BUSCHELMAN K., EIJKHOUT V., GROPP W. D., KAUSHIK D., KNEPLEY M. G., MCINNES L. C., SMITH B. F., ZHANG H., PETSc Users Manual, Rapport nANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[BEN 99] BENZI M., FROMMER A., NABBEN R., SZYLD D. B., Algebraic theory of Multiplicative Schwarz methods, *Numerische Mathematik*, 1999.

[HAC 99] HACKBUSCH W., *Iterative Solution of Large Sparse Systems of Equations.*, Springer-Verlag, 1999.

[LI 99] SuperLU Version 2,
http://www.nersc.gov/x̃iaoye/SuperLU/, September 1999.

[MEU 99] MEURANT G., *Computer solution of large linear systems*, North Holland, Amsterdam, 1999.

[MUS 05] MUSTAPHA H., DE DREUZY J.-R., ERHEL J., Discrete Fracture Modeling of 3D Flow in Multiscale Fracture Networks, *International Conference on Mathematical and Computational Issues in the Geosciences*, Avignon, SIAM, 2005.

[SAA 03] SAAD Y., *Iterative methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, SIAM, 2003.

[SMI 96] SMITH B. F., BJÖRSTAD P. E., GROPP W. D., *Domain Decomposition Parallel Multilevel Methods for Elliptic Partial Differential Equations.*, Cambridge Press, 1996.