Numerical Experiments with an Overlapping Additive Schwarz Solver for 3-D Parallel Reservoir Simulation

Luca Pavarino
Marcelo Rame

CRPC-TR93373
June 1993

Center for Research on Parallel Computation
Rice University
6100 South Main Street
CRPC - MS 41
Houston, TX 77005

NUMERICAL EXPERIMENTS WITH AN OVERLAPPING ADDITIVE SCHWARZ SOLVER FOR 3-D PARALLEL RESERVOIR SIMULATION

LUCA F. PAVARINO AND MARCELO RAMÉ *

Abstract. Domain decomposition methods are a major area of contemporary research in numerical analysis of partial differential equations. They provide robust, parallel and scalable preconditioned iterative methods for the large linear systems arising when continuous problems are discretized by finite elements, finite differences or spectral methods. This paper presents some numerical experiments on a distributed memory parallel computer, the 512 processor Caltech Touchstone Delta. An overlapping additive Schwarz method is implemented for the mixed finite element discretization of second order elliptic problems in three dimensions, arising from flow models in reservoir simulation. These problems are characterized by large variations in the coefficients of the elliptic operator, often associated with short correlation lengths, which make the problems very ill-conditioned. The results confirm the theoretical bound on the condition number of the iteration operator and show the advantage of domain decomposition preconditioning as opposed to a simpler but less robust diagonal preconditioner.

1. Introduction.

The discretization of elliptic problems arising in reservoir simulation and, more generally, in simulation of flow in porous media, are characterized by large, sparse and very ill-conditioned linear systems. This is mainly caused by rapid and large changes in the flow coefficients of the elliptic operator (in typical rock, changes in permeability of three to five orders of magnitude over a few feet are not uncommon) as well as by the large scale of the problem. In reservoir simulation applications, an elliptic problem must be solved for the pressure field at each time step when the set of model equations is discretized by an IMPES (Implicit-Pressure-Explicit-Saturation) scheme.

In this paper, we solve these linear systems with an overlapping domain decomposition method of additive Schwarz type; see Chan and Mathew [5] and Dryja and Widlund [9] for an introduction to this class of algorithms. Overlapping Schwarz methods has been proven to be very parallelizable, scalable and robust for three dimensional Galerkin problems, see Gropp and Smith [13]. Here, we study the method applied to mixed finite element problems. We refer to Cowsar [6] for the mathematical analysis of this approach and to Ewing and Wang [10] and Mathew [15], [16] for alternative approaches.

We report on several parallel numerical experiments, performed on the 512 processor Intel Delta computer at Caltech. For other experiments with overlapping additive Schwarz methods in two and three dimensions, see Gropp and Smith [13], Skogen [18], Cowsar [6] and Bjørstad and Skogen [2]. For numerical experiments with substructuring (nonoverlapping) methods in three dimensions, see Smith [19], De Roeck and Le Tallec [17] and Mandel and Brezina [14].

This paper is organized as follows. In the next section, we introduce the elliptic problem and its mixed finite element discretization. In Section 3, we first recall the

* Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251. email: pavarino@rice.edu, marcelo@rice.edu. This work was supported by the U.S. Department of Energy under contract DE-FG-05-92ER25142 and by the State of Texas under contract 1059.

definitions and basic results for overlapping additive Schwarz methods and then we introduce the algorithm implemented in this work. Some implementation details are given in Section 4, while the numerical results are presented in Section 5. Conclusions are summarized in Section 6.

2. The Elliptic Problem. Let Ω be a bounded domain in R^3 with piecewise smooth boundary Γ. Given a 3×3 uniformly positive definite matrix $K(x)$ (which represents the flow coefficient tensor) and $f \in L^2(\Omega)$, we consider the elliptic problem

$$\begin{array}{rcll} -\nabla \cdot K(x)\nabla p &=& f & \text{in} \quad \Omega \\ \nabla p \cdot n &=& 0 & \text{on} \quad \Gamma. \end{array} \tag{1}$$

The flow coefficient tensor, as introduced above, is a function of position only. However, in the context of reservoir simulation, the elliptic pressure problem is coupled to one or more time-dependent mass conservation equations, since the coefficient K depends not only on position through the dependence of permeability on position, but also on phase saturation (i.e., fraction of total pore volume occupied by a phase) or concentration, which are themselves functions of both position and time. The IMPES formulation of the discrete problem breaks the coupling by evaluating the coefficient of the elliptic problem at the previous time level. Because of this, the flow coefficient (also known as total or phase mobility) is assumed to be a function of position only.

Introducing the new vector unknown $\mathbf{u} = -K(x)\nabla p$, we can write (1) as

$$\begin{array}{rcll} K^{-1}\mathbf{u} &=& -\nabla p & \text{in} \quad \Omega \\ \nabla \cdot \mathbf{u} &=& f & \text{in} \quad \Omega \\ \nabla p \cdot n &=& 0 & \text{on} \quad \Gamma. \end{array} \tag{2}$$

The variational formulation of this problem consists in finding $(\mathbf{u}, p) \in H_0(\text{div};\Omega) \times L^2(\Omega)$ such that

$$\int_\Omega K^{-1}\mathbf{u} \cdot \mathbf{v} dx - \int_\Omega p \nabla \cdot \mathbf{v} dx = 0, \qquad \forall \mathbf{v} \in H_0(\text{div};\Omega), \tag{3}$$

$$\int_\Omega \nabla \cdot \mathbf{u} q dx = \int_\Omega f q dx, \qquad \forall q \in L^2(\Omega), \tag{4}$$

where $H_0(\text{div};\Omega)$ is the kernel of the normal trace mapping of $H(\text{div};\Omega)$ into $L^2(\Gamma)$. In order to discretize (3) and (4), we introduce a triangulation $\mathcal{T}_h$ of Ω into elements of size h satisfying the usual regularity requirements. We also consider a coarse triangulation of Ω into nonoverlapping subdomains Ω_i of size H, each consisting of a union of elements of $\mathcal{T}_h$. A standard mixed finite element approximation of (3) and (4) is obtained by introducing finite dimensional subspaces $W_h(\Omega) \subset H_0(\text{div};\Omega)$ and $V_h(\Omega) \subset L^2(\Omega)$ associated to the triangulation $\mathcal{T}_h$: find $(\mathbf{u}_h, p_h) \in W_h(\Omega) \times V_h(\Omega)$ such that

$$\int_\Omega K^{-1}\mathbf{u}_h \cdot \mathbf{v}_h dx - \int_\Omega p_h \nabla \cdot \mathbf{v}_h dx = 0, \qquad \forall \mathbf{v}_h \in W_h(\Omega), \tag{5}$$

$$\int_\Omega \nabla \cdot \mathbf{u}_h q_h dx = \int_\Omega f q_h dx, \qquad \forall q_h \in V_h(\Omega). \tag{6}$$

We assume that this problem is well posed, i.e. that the Babuska-Brezzi inf-sup condition holds; see Brezzi [3]. For a review of the most important mixed finite element spaces, see Brezzi and Fortin [4]. We eliminate the velocity unknown $\mathbf{u}_h$ by introducing a discrete gradient operator $\nabla_h : V_h(\Omega) \to W_h(\Omega)$ such that

$$\int_\Omega K^{-1}(\nabla_h[q_h]) \cdot \mathbf{v}_h dx = -\int_\Omega q_h \nabla \cdot \mathbf{v}_h dx, \qquad \forall \mathbf{v}_h \in W_h(\Omega). \tag{7}$$

Problem (5) and (6) is then equivalent to the problem for the pressure unknown p_h

$$a(p_h, q_h) = \int_\Omega f q_h dx, \qquad \forall q_h \in V_h(\Omega), \tag{8}$$

where

$$a(p_h, q_h) = \int_\Omega K^{-1} \nabla_h[p_h] \nabla_h[q_h] dx = -\int_\Omega q_h \nabla \cdot (\nabla_h[p_h]) dx.$$

We recall that by choosing a basis in $V_h(\Omega)$ and computing the integrals in (7) by special quadrature rules, the resulting linear system

$$Ax = b \tag{9}$$

is equivalent to a cell-centered finite difference discretization (see Weiser and Wheeler [20]). We solve this linear system by an iterative method using an overlapping domain decomposition preconditioner of additive Schwarz type.

3. Additive Schwarz Methods.

3.1. Abstract theory.

For a detailed presentation of additive and multiplicative Schwarz methods, see Dryja, Smith and Widlund [8] and the references therein. Let V be a finite dimensional Hilbert space and $a(\cdot,\cdot) : V \times V \to R$ a selfadjoint, elliptic and bounded bilinear form. We are solving the problem: find $u \in V$ such that

$$a(u, v) = f(v), \qquad \forall v \in V. \tag{10}$$

The space V can be decomposed into a (not necessarily direct) sum of $N+1$ subspaces

$$V = V_0 + V_1 + \cdots + V_N,$$

where the first space V_0 is related to a special coarse discretization of the problem. Let $b_i(\cdot,\cdot) : V_i \times V_i \to R$ be inner products and $T_i : V \to V_i$ be operators defined by

$$b_i(T_i u, v) = a(u, v), \qquad \forall v \in V_i.$$

If $b_i(u, v) = a(u, v)$, then $T_i = P_i$, the a-orthogonal projection onto V_i. This choice corresponds to the use of exact solvers for the problems on the subspaces $V_i, i = 0, 1, \cdots, N$. The additive Schwarz operator is defined as

$$T = T_0 + T_1 + \cdots + T_N.$$

The original problem (10) is replaced by the preconditioned problem

$$Tu = g, \tag{11}$$

where the right hand side $g = \sum_{i=0}^{N} g_i = \sum_{i=0}^{N} T_i u$ is constructed by solving

$$b_i(g_i, v) = a(u, v) = f(v), \qquad \forall v \in V_i.$$

This linear system is solved iteratively and the iteration is usually accelerated by a Krylov space method: if the original problem is symmetric and positive definite, we can use the conjugate gradient method, otherwise we can use GMRES or other methods for general systems; see Freund, Golub and Nachtigal [11] for an introduction to Krylov methods. The applications where (10) is the discretization of an elliptic problem over a domain Ω decomposed into subdomains Ω_i is particularly important. In the classical overlapping additive Schwarz method, each Ω_i is extended a certain number of elements beyond the boundary to a larger subdomain Ω_i'. The number of elements used in this extension determines the overlap of the new decomposition $\{\Omega_i'\}$, which is measured by the distance δ between the boundaries of Ω_i and Ω_i'. The space decomposition in the Galerkin finite element formulation is given by

- $V_0 =$ space of trilinear basis functions defined on the coarse mesh and satisfying the given boundary conditions;
- $V_i = V_h \bigcap H_0^1(\Omega_i'), i = 1, \cdots, N$.

Dryja and Widlund [9] proved that the iteration operator P defined by these subspaces satisfies:

THEOREM 3.1. *When exact solvers are used for the subproblems, the condition number of the additive Schwarz method satisfies*

$$\kappa(P) \leq C(1 + \frac{H}{\delta}),$$

where C is independent of H, h and δ. For more general formulations of this result, see Dryja and Widlund [9]. This result has been extended by Cowsar [6] to show that the same bound holds true for certain mixed and hybrid finite element methods.

3.2. The algorithm. We now describe an additive Schwarz method with minimal overlap for the discrete problem (9). For each subdomain Ω_i, let Ω_i' be the extension with minimal overlap ($\delta = h$). This choice of minimal overlap is motivated by several numerical results in the current literature, which have shown that minimal overlap often gives optimal timings (see Dryja and Widlund [9] and the references therein). The coarse and local subspaces defining the algorithm are

- $V_0 = \{\phi \in V_h | \phi =$ interpolant at the V_h nodes of the trilinear basis functions of $V_H\}$;
- $V_i = \{\phi \in V_h | supp(\phi) \subset \Omega_i'\}$,

respectively. Exact direct solvers are used for both the local and coarse problems. We note that the matrix representation of the operator P defined by these subspaces is given by $P = B^{-1}A$, where

$$B^{-1} = R_H^T A_H^{-1} R_H + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i. \tag{12}$$

Here R_i is a restriction matrix that maps global vectors to local vectors in V_i, R_H has the same structure as the two level multigrid restriction operator, $A_i = R_i A R_i^T$ and $A_H = R_H A R_H^T$ (see (Chan and Mathew, 1994)). Applying Cowsar's extension of Theorem 3.1 (Cowsar, 1993) to our algorithm, we obtain a condition number bounded by $C(1 + \frac{H}{h})$, with a constant C independent of H and h.

4. Parallel implementation. We have implemented this overlapping additive Schwarz solver on several distributed memory MIMD computers. In this paper, we report on results with the Caltech Touchstone Delta, which has 512 Intel i860 processing nodes, each with 16M bytes of local memory, interconnected in a two-dimensional mesh.

A single-program-multiple-data (SPMD) programming model is used for our Fortran 77 implementation. Each (overlapping) subdomain is assigned to one processor. In the preconditioned conjugate gradient method running on each processor, the four major computational kernels are vector updates, inner products, matrix-vector multiplies and preconditioning. The vector updates and the inner products are easily parallelized; the latter require communication for the global sum of the local inner products.

The coefficient K in equation (9) is a diagonal tensor throughout the experiments presented in this work, which allows for a discretization of the problem equivalent to that of a 7 point cell-centered finite difference stencil. The resulting stiffness matrix has only 7 nonzero diagonals which can be stored in local one-dimensional arrays. Therefore the matrix-vector products required in the conjugate gradient iteration are easily parallelized. Extensions of this implementation to handle a full tensor K are in preparation.

The additive Schwarz preconditioner is naturally parallel to a large extent, since the local subdomain problems are not coupled and can therefore be solved simultaneously. The only nontrivial step concerns the parallel processing of the coarse problem. In our implementation, we use the simplest choice on MIMD architectures, also adopted by (Smith, 1993): we construct the coarse matrix in parallel, but we store and solve the entire coarse problem on all of the processors. For a large number of subdomains, this is not a good choice, since the coarse problem will then dominate the total cost. In that case, we should use a more elaborate implementation of the algorithm and solve the coarse problem in parallel or in a multilevel scheme. In our experiments, the cost of solving the coarse problem was always a low percentage of the total cost. See Bjørstad and Skogen [2] for a multilevel implementation on a SIMD computer with thousands of processors.

For a discussion of parallelism in domain decomposition, see Gropp [12]. For a general discussion of parallelism in iterative and direct methods, see Demmel, Heath, and Van der Vorst [7].

5. Numerical experiments. We present here the results of numerical experiments where the discrete problem (9) is solved iteratively using a preconditioned conjugate gradient method with preconditioner B^{-1} and zero initial guess. The coefficient

K is a diagonal tensor throughout the experiments presented in this work. B^{-1} is given either by the overlapping additive Schwarz method (ASM) introduced in the previous sections, or by a simple diagonal scaling also known as Jacobi method (JCG). Since the two preconditioned problems can have eigenvalues of extremely different order of magnitude, we have chosen to stop the iteration when

$$\kappa(B^{-1}A)\sqrt{\frac{(r, B^{-1}r)}{(b, B^{-1}b)}} < 10^{-6}.$$

Here r is the current residual and $\kappa(B^{-1}A)$ is the condition number of the iteration operator, easily computed by the Lanczos recursion in the conjugate gradient. This stopping criterion is preconditioner independent and guarantees roughly a six-digit precision in the energy norm. See Ashby, Manteuffel and Saylor [1] for a discussion of stopping criteria.

The local and coarse problems in the ASM preconditioner are solved directly with Lapack banded subroutines, which are expensive but very robust. We are in the process of studying the use of approximate local solvers. Comparison of exact and approximate solvers can be found in Skogen [18] for SIMD computers and in Smith (1993) for MIMD computers.

Scaling studies in domain decomposition are usually conducted by keeping the global domain size fixed and increasing the number of processors by decreasing the subdomain size H/h. In this way, the algorithm is changed for every choice of subdomain size and it is difficult to discern among the multiple effects of changes in subdomain size, aspect ratio and surface to volume ratio effects, all of which happen concurrently as the number of subdomains (processors) is changed. Moreover, because of the restricted computing memory per processor, only a problem of modest global size can be run if one is to normalize speed-ups with respect to the single-processor run.

Another type of scaling consists of keeping the subdomain size H/h fixed and increasing the number of processors by increasing the number of subdomains and therefore increasing the size of the global discrete problem. We have chosen this type of scaling in the first two sets of experiments, described in Subsection 5.1 amd 5.2. This is equivalent to refining the discretization by using more subdomains with fixed $\frac{H}{h}$ on a problem of constant global size. In this case, however, it should be noted that because the flow coefficients are assigned on a per-subdomain basis, the correlation lengths of the coefficient distribution decrease as the problem is solved on more subdomains (processors). This means that we are not solving the same exact problem (problems indeed become more difficult) as more subdomains are added, but the contrived test cases shown below distinctively point to some features of the domain decomposition solver implemented in this work. For example, we can check the constant bound predicted by Theorem 3.1 and the actual size of the constant by just having the coefficient jumps aligned with subdomain interfaces but irrespective of the actual correlation lengths of the coefficient heterogeneity.

When the matrix K is the identity (Poisson equation), diagonal preconditioning is definitely faster for the presented problem sizes, but other test cases presented below, which involve a jumping and/or anisotropic tensor K, show the robustness of the domain decomposition solver implemented in this work.

We always consider a solid, rectangular domain decomposed into solid, rectangular subdomains. A subdomain size is measured by the number of degrees of freedom (elements) in each coordinate direction: for example, a subdomain of size $10 \times 10 \times 10$ has 10 degrees of freedom in each direction.

Each case was run several times to minimize the timing variations due to factors like machine load, numbers of users, etc.. We have observed variations of a few percent only in the communication timings and we report here the minimum in each case. In our tables and Figures, κ denotes the condition number of the iteration operator and n_{it} the number of iterations.

5.1. Planar scaling. We first consider a "flat" domain Ω which is only one subdomain thick in the z-direction and $2n \times 2n$ subdomains in the xy-plane (see Figure 1). Each subdomain has size $10 \times 10 \times 10$ and we scale the decomposition as $(2,2,1),(4,4,1),\cdots,(16,16,1)$ subdomains. The global problem size varies from $4,000$ to $256,000$ unknowns. The tensor K is diagonal and constant on each subdomain, but varying four orders of magnitude across subdomains and anisotropic. As shown in Figure 1, we "slice" the domain in the x-direction and define $K = I$ in one slice and $diag(K) = (10^3, 10^{-1}, 10)$ in the next. The results and timings are reported in Table 1 and 2, and plotted in Figure 5, 6, 7, 8. In all tables, *total* is the total time for the method to converge to the desired precision. For the ASM results of Table 1, *total* is the sum of initialization and iteration times; we report only the latter (*iter*), so the initialization time can be obtained by difference. The part of *iter* spent in the preconditioning step, i.e. solving local and coarse problems, is reported in the last two columns, *local* and *coarse* (*coarse* is also reported as a percentage of *total*). In the JCG results of Table 2, there is virtually no initialization time, so here *total* $\approx$ *iter* is split into the time for preconditioning (*prec.*), matrix multiplication (*A-mult*) and inner product (*lip* = local inner product and *psum* = parallel global sum). We have not timed the vector updates, since they are completely parallel, as are the local inner products. Discussion of both the planar and the cubic scalings is given in the next subsection.

5.2. Cubic scaling. We now consider a domain Ω which has $2n$ subdomains in the x-direction and $n \times n$ subdomains in the yz-plane (see Figure 2). Each subdomain has size $13 \times 13 \times 13$ and we scale the decomposition as $(2,1,1),(4,2,2),\cdots,(10,5,5)$ subdomains. The global size varies from $4,394$ to $549,250$ unknowns. The tensor K is as in the previous planar case. The results are reported in Table 3 and 4, and plotted in Figure 9, 10, 11, 12. The timed quantities are the same as the previous case.

It is clear from the results that JCG does not scale well, while ASM scales well up to the point where the assembling of the global stiffness coarse-grid matrix and solution of the coarse problem account for a significant percentage of the total execution time . In the planar case, this point is around 100 subdomains; in the cubic case, this point is not yet reached, due to the larger subdomain size. The reason for the poor performance of JCG is that the condition number grows like $1/h^2$ (and therefore the number of iteration grows like $1/h$), while for ASM it only grows like H/h, which is kept constant.

Therefore, most of the JCG time is spent iterating, doing matrix multiplies and global sums for the local inner products, since the preconditioner is completely local (Figure 8 and 12). On the contrary, the initialization time for ASM is considerable, due to the expensive exact factorization of the local problems and to the parallel construction of the coarse problem, which involves communication. Consequently, the iteration time for ASM is mostly spent on the preconditioning step (Figure 7 and 11). Note the almost ideal curve for the local part of the ASM preconditioner, while the cost of the coarse problem increases with the number of subdomains. We have also run ASM without a coarse space in the planar case with 256 subdomains: the condition number increased to $2,737$ and the number of iterations to 138. Therefore the coarse problem is essential to produce a scalable algorithm.

5.3. 3-D checkerboard. We now study how the magnitude of the coefficient jumps affects the methods. The global domain $\Omega = [0,1]^3$ is decomposed into $64 = 4 \times 4 \times 4$ cubic subdomains, each with size $13 \times 13 \times 13$. The coefficient jumps are isotropic and arranged in a checkerboard fashion as in Figure 3: in the white subdomains $K = I$, while in the shaded subdomains $K = 10^{\alpha} I$. The results, reported in Table 5, show the rapid deterioration of the JCG condition number, which grows as $10^{\alpha+2}$ and the number of iterations, which grows linearly in α. It should be noted that this result is within the bound $n_{it} \le C\sqrt{\kappa}$. In the cases marked with a $*$ symbol, JCG failed to converge. On the other hand, after an initial growth, ASM becomes insensitive to the coefficient jumps and the condition number seems to converge to a limit.

We have also run a large anisotropic problem with 1,124,864 unknowns, where a cubic domain $\Omega = [0,1]^3$ is divided into $512 = 8 \times 8 \times 8$ subdomains of size $13 \times 13 \times 13$. The coefficient jumps are again arranged in a checkerboard fashion, but with anisotropic values as in the case of cubic scaling. Table 6 shows the results for ASM. In this case, JCG did not converge in 10,000 iterations.

5.4. Unaligned coefficient jumps. In the previous three cases, the coefficient jumps were always aligned with the subdomain boundaries. In order to study the performance of the methods when applied to more realistic problems, we consider coefficient jumps which are not aligned with the subdomain boundaries. We have two cases: a) the jumps are in the subdomains interior (no jumps on the interface) or b) the jumps are across subdomain boundaries (jumps on the interface). In a cubic domain $\Omega = [0,1]^3$, the coefficient K equals $10^{\alpha} I$ inside two spheres of radius 0.2 centered at $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ and $(\frac{3}{4}, \frac{3}{4}, \frac{3}{4})$, while $K = I$ in the rest of Ω; see Figure 4. Ω is discretized by $24 \times 24 \times 24$ elements. In Table 7, we report the results obtained by increasing the exponent α. The $2 \times 2 \times 2$ decomposition corresponds to case a), because each sphere is completely contained in a subdomain (column 2 and 3; $H/h = 12$). The $4 \times 4 \times 4$ decomposition corresponds to case b), because now the subdomain boundaries cut across the spheres (column 4 and 5; $H/h = 6$). The JCG results are reported in column 6 and 7 of the same table, and show the same rapid growth of κ and the number of iterations as α increases, as in the checkerboard case. It is clear that in case a), ASM becomes insensitive to α after an initial moderate growth. On the other hand, in case b) the rate of convergence of ASM deteriorates after an initial slow growth: κ starts growing as $10^{\alpha-3}$ and the number of iterations grows linearly with α. Therefore, the presence of coefficient jumps which cut across the subdomain boundaries seriously deteriorates the convergence rate of ASM. Similar results have been reported for a nonoverlapping method (see Mandel and Brezina [14]), where the deterioration of the convergence rate seems even more pronounced.

In the last Table 8, we report a traditional scaling for a problem with jumps unaligned with the subdomain boundaries, which gives more insight into the causes of deterioration of the ASM convergence rate. The domain $\Omega = [0,1]^3$ is again as in Figure 4, but the coefficient K is anisotropic: $diag(K) = (10^2, 10^{-1}, 10)$ inside the two spheres and $K = I$ in the rest of Ω. We fix a small global size, $24 \times 24 \times 24$, in order to start the scaling with a small number of subdomains. We then decompose Ω into an increasing number of subdomains of smaller size. For this problem, JCG converges in 381 iterations with a condition number $\kappa = 5,378$ (of course independent of the decomposition). Now multiple effects occur at the same time for ASM: N and H/h change in each decomposition; some decompositions have cubic subdomains, some have solid rectangular subdomains, therefore the subdomains aspect ratio change; the coefficient jumps cut across the subdomain interfaces, except in the first decomposition with $8 = 2 \times 2 \times 2$ subdomains, which has therefore the only favorable condition number. The reported timings show the adverse effect of the unaligned coefficient jumps, both in the tangential and normal direction to the subdomain faces (in the decomposition with 27 and 64 subdomains, for which the subdomain aspect ratio is the same as that of the 8 subdomain decomposition) and of changing aspect ratio (in the decompositions with 16 and 32 subdomains), in addition to the unaligned coefficient jumps.

6. Conclusions. A parallel implementation of an additive Schwarz preconditioner for the conjugate gradient method is presented in this work, in reference to the elliptic pressure problem arising from the IMPES formulation of the discrete reservoir model equations. Several numerical experiments were conducted to investigate the robustness of the domain decomposition solver and to verify theoretical bounds on the condition number and the iteration count.

The results for flow coefficient distributions where jumps are aligned with the subdomain interfaces show that JCG does not scale well, while ASM scales well up to the point where the assembling of the global stiffness coarse-grid matrix and solution of the coarse problem account for a significant percentage of the total execution time. The reason for the poor performance of JCG is that the condition number grows like $1/h^2$ (and therefore the number of iteration grows like $1/h$), while for ASM it only grows like H/h. Most of the JCG time is spent iterating, doing matrix-vector products and global sums for the local inner products, since the preconditioner is completely local. On the other hand, the initialization time for ASM is considerable, due to the expensive exact factorization of the local problems and to the parallel construction of the coarse problem, which involves communication. Moreover, the iteration time for ASM is spent mostly in the preconditioning step. Running ASM without a coarse space produced condition numbers and iteration counts that grow with the number of subdomains, showing that the coarse problem is essential to produce a scalable algorithm.

Tests were conducted to investigate the effect of growing jump discontinuities in the flow coefficient on the performances of both JCG and ASM. The results show that the JCG condition number and iteration count grow as the magnitude of the discontinuity increases, regardless of the location of the discontinuities. On the other hand, after an initial growth, ASM becames insensitive to the magnitude of the coefficient jumps, if these are aligned with the subdomain boundaries or unaligned but contained in the subdomain interiors. If the coefficient jumps are unaligned with the subdomain boundaries and cut across these boundaries, then the convergence rate for ASM deteriorates.

These results are valid also for anisotropic coefficients, which seem to make the discrete problem particularly difficult for JCG.

Further studies are being conducted for large heterogeneous problems, the use of inexact solvers for the subdomain problems, the implementation of alternative coarse spaces and the comparison of different parallel architectures. Future directions of research will include the extension of the algorithm to nonsymmetric and indefinite problems, to a full tensor K and to systems of equations.

Acknowledgments. We would like to thank Frédéric d'Hennezel and Lawrence Cowsar for helpful discussions.

REFERENCES

[1] S. Ashby, T. A. Manteuffel, and P. E. Saylor, *A taxonomy for conjugate gradient methods*, SIAM J. Numer. Anal., (1990), pp. 1542–1568.

[2] P. E. Bjørstad and M. Skogen, *Domain decomposition algorithms of Schwarz type, designed for massively parallel computers*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, D. E. Keyes, G. A. Meurant, J. S. Scroggs, and R. G. Voigt, eds., Philadelphia, PA, 1992, SIAM.

[3] F. Brezzi, *On the existence, uniqueness and approximation of the saddle-point problems arising from lagrangian multipliers*, RAIRO Anal. Numer., (1974), pp. 129–151.

[4] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer–Verlag, 1991.

[5] T. F. Chan and T. P. Mathew, *Domain decomposition algorithms*, Acta Numerica (1994), Cambridge University Press, 1994, pp. 61–143.

[6] L. C. Cowsar, *Dual variable Schwarz methods for mixed finite elements*, Tech. Rep. TR93-09, Department of Mathematical Sciences, Rice University, March 1993.

[7] J. W. Demmel, M. T. Heath, and A. V. D. Vorst, *Parallel numerical linear algebra*, Acta Numerica (1993), (1993), pp. 111–197.

[8] M. Dryja, B. F. Smith, and O. B. Widlund, *Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions*, Tech. Rep. 638, Department of Computer Science, Courant Institute, May 1993. To appear in SIAM J. Numer. Anal.

[9] M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci. Comput., 15 (1994), pp. 604–620.

[10] R. E. Ewing and J. Wang, *Analysis of the Schwarz algorithm for mixed finite element methods*, RAIRO Model. Math. Anal. Numer., (1992), pp. 739–756.

[11] R. Freund, G. H. Golub, and N. Nachtigal, *Iterative Solution of Linear Systems*, Acta Numerica (1992), Cambridge University Press, 1992, pp. 57–100.

[12] W. D. Gropp, *Parallel computing and domain decomposition*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. F. Chan, D. E. Keyes, G. A. Meurant, J. S. Scroggs, and R. G. Voigt, eds., Philadelphia, PA, 1992, SIAM.

[13] W. D. Gropp and B. F. Smith, *Experiences with domain decomposition in three dimensions: Overlapping Schwarz methods*, in Domain Decomposition Methods in Science and En-

gineering: The Sixth International Conference on Domain Decomposition, Y. A. Kuznetsov, J. Périaux, A. Quarteroni, and O. B. Widlund, eds., vol. 157, AMS, 1994. Held in Como, Italy, June 15–19,1992.

[14] J. Mandel and M. Brezina, *Balancing domain decomposition: Theory and computations in two and three dimensions*, tech. rep., Computational Mathematics Group, University of Colorado at Denver, 1992. Submitted to Math. Comp.

[15] T. P. Mathew, *Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems, part I: Algorithms and Numerical results*, Numer. Math., 64 (4) (1993), pp. 445–468.

[16] ———, *Schwarz alternating and iterative refinement methods for mixed formulations of elliptic problems, part II: Theory*, Numer. Math., 64 (4) (1993), pp. 469–492.

[17] Y.-H. D. Roeck and P. L. Tallec, *Analysis and test of a local domain decomposition preconditioner*, in Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. Kuznetsov, G. Meurant, J. Périaux, and O. B. Widlund, eds., SIAM, Philadelphia, PA, 1991.

[18] M. D. Skogen, *Parallel Schwarz Methods*, PhD thesis, Department of Informatics, University of Bergen, Norway, February 1992.

[19] B. F. Smith, *A parallel implementation of an iterative substructuring algorithm for problems in three dimensions*, SIAM J. Sci. Comput., 14 (1993), pp. 406–423.

[20] A. Weiser and M. F. Wheeler, *On convergence of block-centered finite difference for elliptic problems*, SIAM J. Numer. Anal., (1988), pp. 351–357.

Table 1
ASM planar scaling, $H/h = 10$

N and decomposition	κ	n_{it}	Timings (in secs.)				
			total	iter.	local	coarse	(% of total)
4=2×2×1	9.611	19	4.88	1.90	1.48	0.22	(4.51)
16=4×4×1	13.476	25	7.55	3.38	2.55	0.42	(5.56)
36=6×6×1	14.694	26	7.73	3.54	2.63	0.50	(6.47)
64=8×8×1	15.132	26	7.90	3.65	2.65	0.58	(7.34)
100=10×10×1	15.015	27	8.41	4.06	2.83	0.74	(8.80)
144=12×12×1	14.765	26	8.51	4.02	2.68	0.88	(10.34)
196=14×14×1	14.771	26	9.02	4.29	2.70	1.11	(12.31)
256=16×16×1	14.7706	26	9.69	4.54	2.70	1.36	(14.03)

Table 2
JCG planar scaling, $H/h = 10$

N and decomposition	κ	n_{it}	Timings (in secs.)				
			total	prec.	A-mult	psum	lip
4=2×2×1	216,795	1,683	15.04	0.78	8.11	2.73	1.61
16=4×4×1	632,935	3,570	37.54	1.78	20.52	11.86	3.96
36=6×6×1	$1.3{\cdot}10^6$	5,318	58.94	2.63	30.50	20.53	5.91
64=8×8×1	$2.3{\cdot}10^6$	6,769	76.81	3.43	39.02	27.76	7.50
100=10×10×1	$3.4{\cdot}10^6$	8,186	99.55	4.12	48.97	39.94	9.10
144=12×12×1	$> 5 \cdot 10^6$	9,715	119.32	4.85	59.03	49.15	11.59
196=14×14×1	$> 5 \cdot 10^6$	>10,000	*	*	*	*	*
256=16×16×1	$> 5 \cdot 10^6$	>10,000	*	*	*	*	*

Table 3
ASM cubic scaling, $H/h = 13$

N and decomposition	κ	n_{it}	Timings (in secs.)				
			total	iter.	local	coarse	(% of total)
2=2×1×1	4.97	14	16.22	3.64	3.17	0.27	(1.66)
16=4×2×2	79.59	52	38.83	19.55	16.47	1.64	(4.22)
54=6×3×3	115.58	65	61.27	32.02	27.18	2.87	(4.68)
128=8×4×4	138.71	73	66.31	36.84	30.71	3.70	(5.58)
250=10×5×5	153.94	78	71.51	41.45	33.19	5.25	(7.34)

Table 4
JCG cubic scaling, $H/h = 13$

N and decomposition	κ	n_{it}	Timings (in secs.)				
			total	prec.	A-mult	psum	lip
2=2×1×1	$1.5\cdot 10^5$	1,552	21.38	1.53	11.58	1.54	2.89
16=4×2×2	$3.7\cdot 10^5$	3,786	75.03	4.25	43.12	12.30	8.53
54=6×3×3	$6.7\cdot 10^5$	5,911	136.98	7.52	80.00	35.96	16.26
128=8×4×4	$1.1\cdot 10^6$	7,652	183.89	9.74	104.73	55.81	20.81
250=10×5×5	$1.8\cdot 10^6$	9,233	281.28	11.81	143.08	120.68	24.05

Table 5
4×4×4 checkerboard with increasing coefficient jumps, $H/h = 13$

	ASM		JCG	
10^α	κ	n_{it}	κ	n_{it}
10^0	18.4165	24	3,204	247
10^1	19.4158	26	5,605	377
10^2	52.1034	36	30,057	455
10^3	77.0733	38	$2.73\cdot 10^5$	532
10^4	81.3222	37	$2.71\cdot 10^6$	613
10^5	81.7771	35	$2.70\cdot 10^7$	688
10^6	81.8229	35	$2.70\cdot 10^8$	763
10^7	81.8275	35	*	*
10^8	81.8280	34	*	*
10^9	81.8276	36	*	*
10^{10}	81.8282	34	*	*

Table 6
8×8×8 checkerboard, $H/h = 13$, global size 1,124,864

	N	κ	n_{it}	Timings (in secs.)				
ASM				total	iter.	local	coarse	(% of total)
	512=8×8×8	241.66	101	94.217	60.37	43.63	12.19	(12.94)
JCG	512=8×8×8	did not converge in 10,000 iterations						

Table 7
Increasing coefficient jumps unaligned with subdomain boundaries

	ASM				JCG	
	2×2×2		4×4×4			
10^α	κ	n_{it}	κ	n_{it}	κ	n_{it}
10^0	13.68	17	10.44	20	$6.6 \cdot 10^2$	110
10^1	14.26	17	16.17	23	$1.6 \cdot 10^3$	114
10^2	17.54	18	29.59	26	$1.1 \cdot 10^4$	126
10^3	25.57	19	41.48	28	$1.1 \cdot 10^5$	137
10^4	34.81	19	100.56	31	$1.1 \cdot 10^6$	147
10^5	36.88	19	723.81	35	$1.1 \cdot 10^7$	158
10^6	37.12	19	6,972.2	39	$1.1 \cdot 10^8$	168

Table 8
ASM, jumps unaligned with subdomain boundaries, fixed global size 24×24×24

N and				Timings (in secs.)			
decomposition	subdomain size	κ	n_{it}	total	iter.	local	coarse
8=2×2×2	12×12×12	16.95	23	17.86	5.86	4.85	0.53
16=4×2×2	6×12×12	212.14	79	12.98	9.65	6.78	1.40
27=3×3×3	8×8×8	87.33	58	7.83	5.53	3.72	0.79
32=4×4×2	6×6×12	193.43	89	7.99	6.34	3.64	1.18
64=4×4×4	6×6×6	134.45	75	5.40	4.31	2.17	0.93

Fig. 1. $1 \times 6 \times 6$ *Planar decomposition*

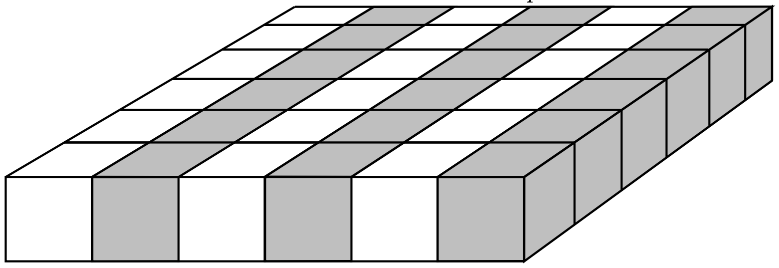


Fig. 2. $6 \times 3 \times 3$ *cubic decomposition*

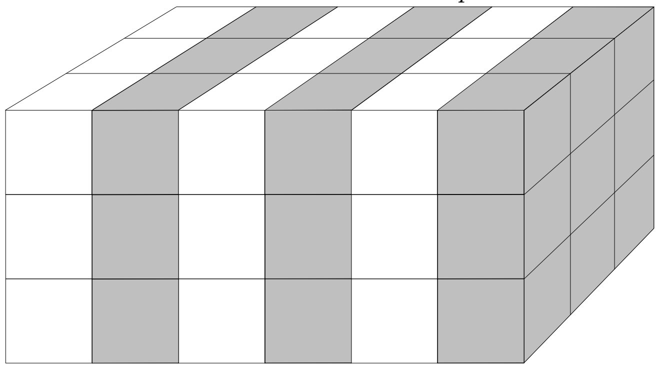


Fig. 3. $4 \times 4 \times 4$ *checkerboard decomposition*

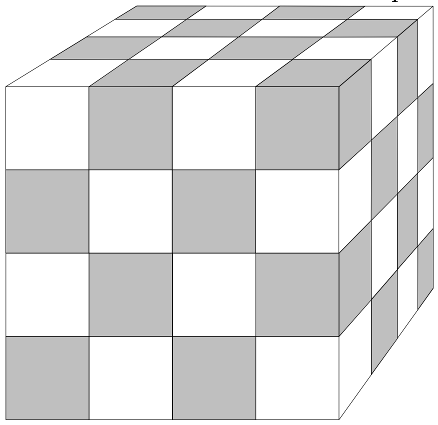


Fig. 4. *coefficient jumps unaligned with subdomain boundaries*

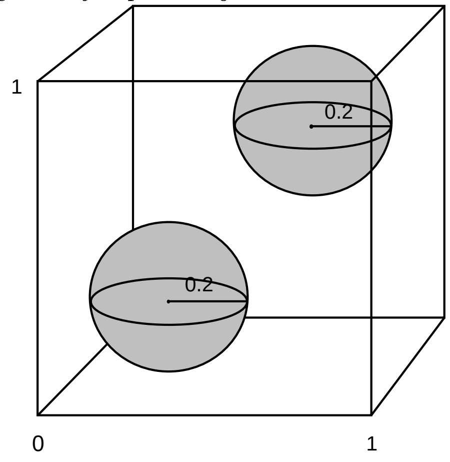


Fig. 5. *ASM and JCG total time*

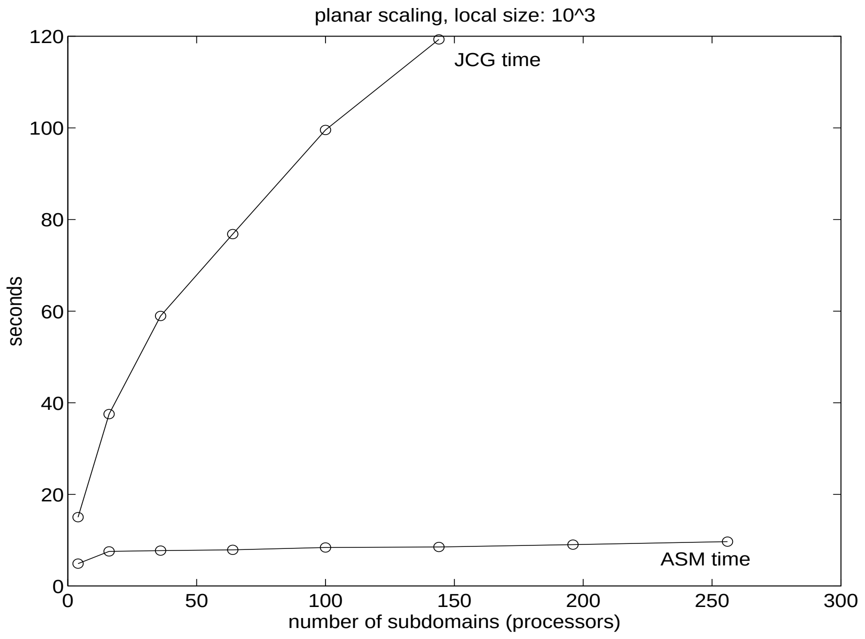


Fig. 6. *Condition number and iteration count for ASM*

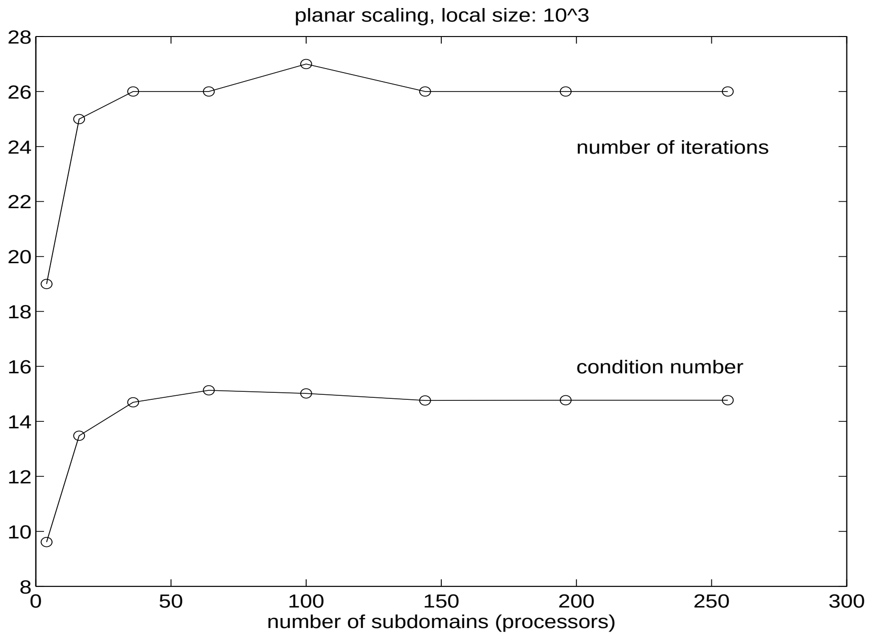


FIG. 7. *Timings for ASM*

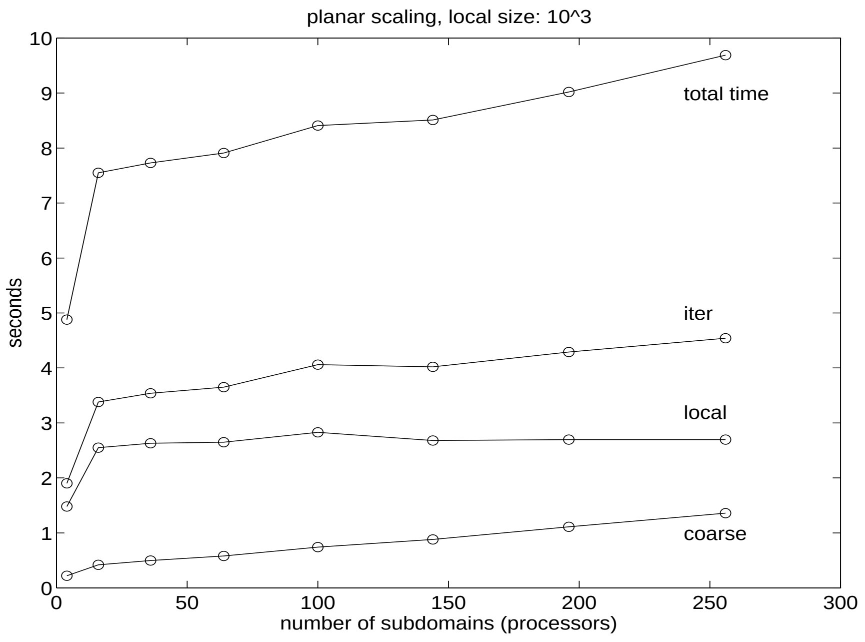


FIG. 8. *Timings for JCG*

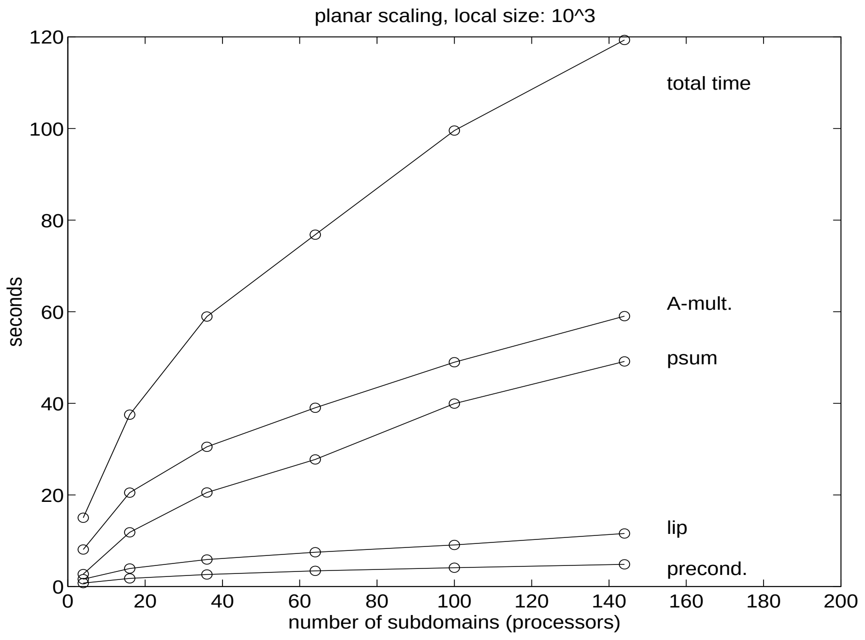


Fig. 9. *ASM and JCG total time*

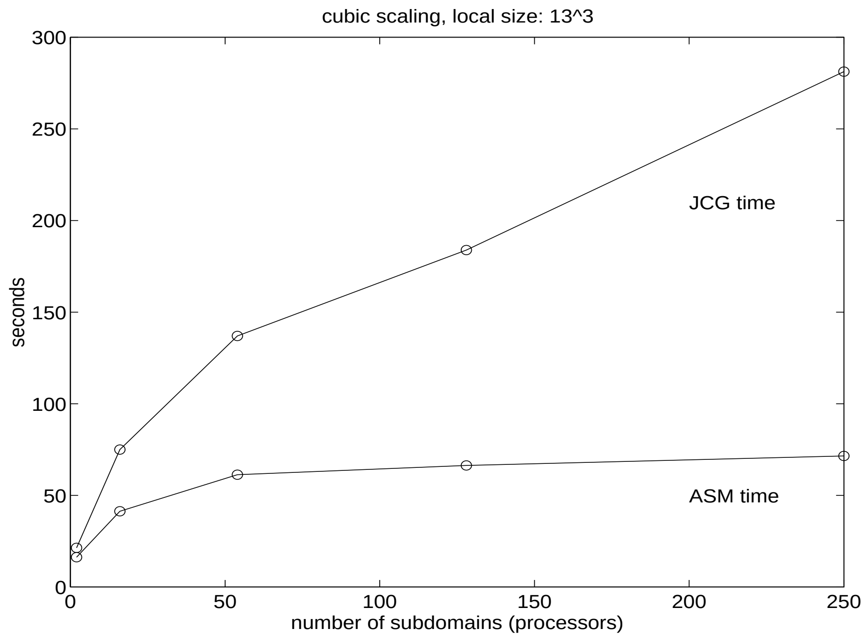


Fig. 10. *Condition number and iteration count for ASM*

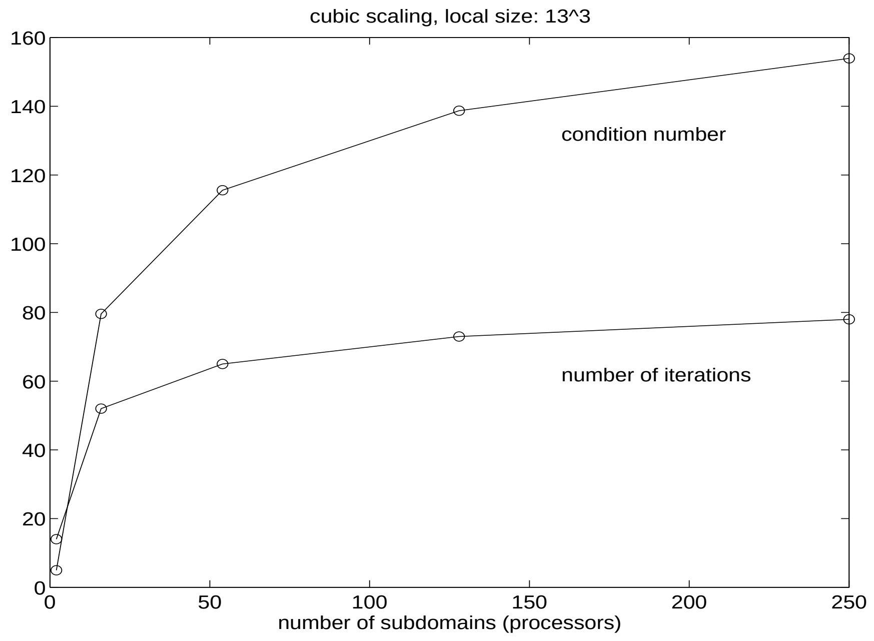


FIG. 11. *Timings for ASM*

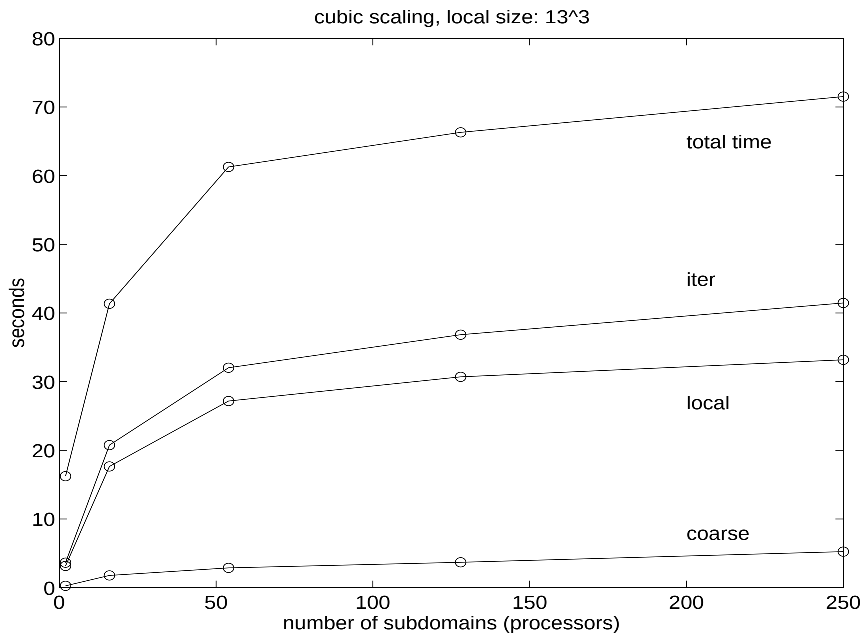
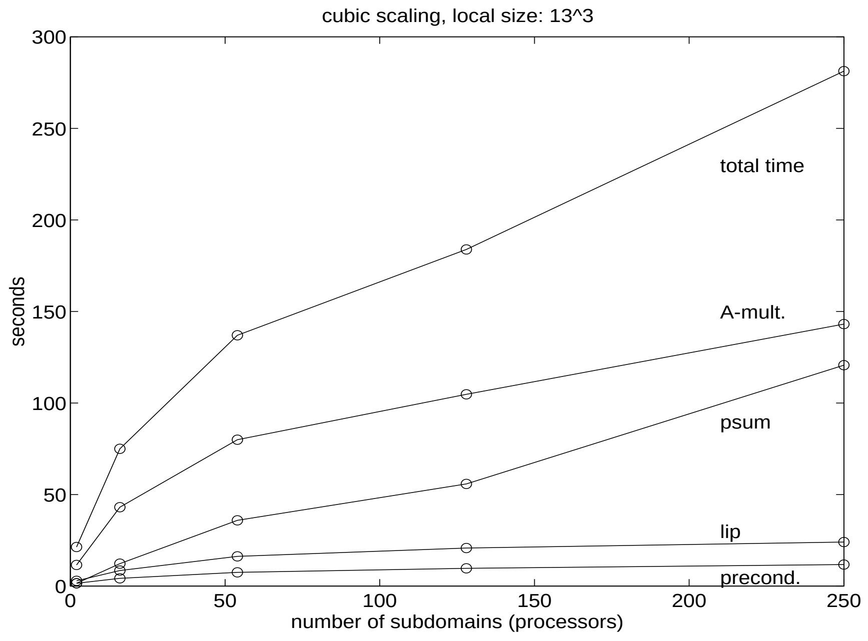


FIG. 12. *Timings for JCG*


cubic scaling, local size: 13^3
seconds
number of subdomains (processors)
total time
A-mult.
psum
lip
precond.