Contemporary Mathematics
Volume 180, 1994

Parallelization of a Multigrid Solver via a Domain Decomposition Method

FRANCOIS-XAVIER ROUX AND DAMIEN TROMEUR-DERVOUT

July 8, 1994

ABSTRACT. This paper reports some experiments with the implementation of a multigrid solver on a distributed memory parallel system. The use of a domain decomposition method for the coarse grid problem is shown to improve the method from both numerical and parallel efficiency points of view. The domain decomposition method is accelerated via a restarting procedure using the direction vectors computed at the previous solutions of the coarse grid problem.

1. Introduction

Multigrid solvers are used in a wide range of applications. The principle of these algorithms is to perform a few iterations of an iterative solver at each grid level, in order to smooth out the components of the residual related to the eigenvectors associated with the highest eigenvalues of the discrete operator and then project it on the coarser grid level. At the coarsest level, the problem is completely solved. Then the error is interpolated back recursively on the finer grid levels. At each level, a few smoothing iterations are performed again. This set of operations defines a cycle of the multigrid algorithm. The complete method consists in iterating cycles to get a good approximation of the solution at the finest grid level. The simplest implementation of this method is made with two grid levels, with V-cycles.

One of the main problems with the parallel implementation of multigrid solvers lies in the fact that the coarse grid problem needs to be solved with a good relative precision. One strategy consists in solving the coarse grid problems with a direct method. But direct methods are difficult to implement efficiently

1991 *Mathematics Subject Classification.* Primary 65M55, 68Q22, 65M06.
This paper is in final form and will not be published elsewhere.

0271-4132/94 $1.00 + $.25 per page

on distributed memory systems, especially for sparse matrices with small bandwidth. Iterative methods are easier to parallelize. But the granularity of the tasks depends upon the number of nodes of the problem, that tends to be small for the coarsest grid levels.

An alternative strategy consists in using a domain decomposition solver for the coarse grid problem, in order to increase the granularity of the tasks. In this paper we present some results with the implementation of a two-grid solver, using a domain decomposition method at the coarse grid level, on a distributed memory machine.

The paper is organized as follows : Section 2 presents some results with the implementation on a 128-processor iPSC-860 computer of a multigrid solver, using a classical iterative procedure at the coarse grid level. Section 3 recalls the principle of the dual Schur complement method. Section 4 presents a restarting technique for the domain decomposition method in order to reduce the number of iterations to be performed for each solution of the coarse grid problem. Section 5 is devoted to a discussion of the intrinsic properties of the dual Schur complement method that makes it a very convenient solver to be used within a multigrid approach, and the conclusions are derived in the last section.

2. Parallel implementation of a two-grid solver

In this section we present the performance of a two-grid method, in order to highlight the difficulties encountered with the parallel efficiency of iterative solvers for the coarse grid problem.

The test problem is a Poisson equation arising from the discretisation by finite difference methods of 3D incompressible Navier-Stokes equations with a velocity-vorticity formulation. The multigrid solver is used for solving the velocity problem at each time step. The fine, respectively coarse, mesh is a 128^3, respectively 64^3, node regular grid. The complete domain is divided into 128 cubic subdomains, and so both grids are decomposed into 128 non-overlapping subgrids. The iterative solver used for the smoothing iterations at the fine grid level and for the solution of the coarse grid problem is Gauss-Seidel for the diagonal blocks associated with the inner nodes of each subdomain, and Jacobi for the off-diagonal blocks associated with the interaction between subdomains.

This solver is naturally parallelizable on a distributed memory machine, each subdomain, with its two-grid, being treated by one processor. At each iteration, only interface nodes are involved in data transfers.

Table 1 presents the results with the implementation of a V-cycle, with 10 smoothing iterations at the fine grid level, and either 200 or 400 iterations for solving the coarse grid problem, on a 128-processor iPSC-860 machine. These results show the degradation of the parallel efficiency due to the decrease of granularity at the coarsest grid level. Of course, the degradation is very sharp here, because the iPSC-860 is a very unbalanced machine: the ratio of the comput-

ing speed over the communication speed is greater than 10 for the present case, according to the time measurements. But on any machine, the time for data transfers should become dominant for some grid levels: if a subdomain consists of a n^3 nodes regular grid, the number of interface nodes is $6n^2$. As the amount of arithmetic operations depends linearly upon the number of nodes within the subdomain, and the amount of data transfers depends upon the number of interface nodes, the ratio of the communication cost over the computational cost would become large on any distributed memory machine for n small enough, which means for grids coarse enough.

Table 1 : Efficiency of multigrid with Gauss-Seidel solver

number of iterations for the coarse grid	efficiency on the fine grid	efficiency on the coarse grid	global efficiency
200	44%	16%	34%
400	44%	16%	27%

3. Solution of the coarse grid problem with the dual Schur complement

The dual Schur complement method for the Poisson equation consists in introducing λ, the flux along the interface between subdomains, and solving iteratively the associated condensed interface problem [1]. The solution of this problem is the λ for which the solution fields of the local Neumann problems are continuous along the interfaces between subdomains. Then, the global field whose restriction to each subdomain is defined as the solution of the local Neumann problem is continuous, and also its normal derivative. It is therefore the solution of the global problem.

In the case of a decomposition in two subdomains, Ω_1 and Ω_2, if A_1 and A_2 are the discretized operators on Ω_1 and Ω_2 with Neumann boundary conditions on the interface Γ_3, and B_1 and B_2 the trace operators over Γ_3 of functions defined upon Ω_1 and Ω_2, the hybrid formulation of the Poisson problem on the domain Ω can be written as follows :

$$\left\{ \begin{array}{rcl} A_1 u_1 + B_1^t \lambda & = & f_1 \\ A_2 u_2 - B_2^t \lambda & = & f_2 \\ B_1 u_1 - B_2 u_2 & = & 0 \end{array} \right. \tag{3.1}$$

By substitution of u_1 and u_2 given by the two first equations in the third one, the condensed interface problem for λ is :

$$(B_1 A_1^{-1} B_1^t + B_2 A_2^{-1} B_2^t)\lambda = B_1 A_1^{-1} f_1 - B_2 A_2^{-1} f_2 \tag{3.2}$$

This problem can be solved by the conjugate gradient algorithm. At each iteration of the condensed interface problem, the local Neumann problems are solved by a direct method, and then the discontinuity of the local solutions along the interfaces are computed. So, the communication costs per iteration are the

same as for the Gauss-Seidel method, because only interface unknowns are to be transferred, but the granularity of the tasks is much larger, because the local task consists of a forward/backward substitution using the dense skyline factorization instead of the sparse initial matrix. Table 2 presents the results with the implementation of a V-cycle, with 10 smoothing iterations at the fine grid level, and either 200 Gauss-Seidel iterations or 20 dual Schur complement iterations for solving the coarse grid problem, on a 128-processor iPSC-860 machine.

Table 2 : Efficiency with Gauss-Seidel or dual Schur complement

coarse grid solver	efficiency on the fine grid	efficiency on the coarse grid	global efficiency
Gauss-Seidel	44%	16%	34%
dual Schur complement	44%	63%	53%

4. Acceleration of the dual Schur complement method

For solving the linear system of equations $Ax = b$, the conjugate gradient algorithm consists in computing a set of direction vectors that are conjugated for the dot product associated with the matrix A, and that generate the Krylov space $\text{Span}\{g_0, Ag_0, \ldots, A^{p-1}g_0\}$, where g_0 is the initial residual $Ax_0 - b$. The approximate solution x_p at iteration p, minimizes the A-norm of the error, $(Ax_p - b, x_p - x)$, over the space $x_0 + \text{Span}\{g_0, Ag_0, \ldots, A^{p-1}g_0\}$.

If a set of conjugate directions (w_i), $1 \leq i \leq p$, is given, then the element x_p of $x_0 + \text{Span}\{w_1, w_2, \ldots, w_p\}$ that minimizes the residual can be easily computed:

$$x_p = x_0 - \sum_{i=1}^{p} \frac{(g_0, w_i)}{(Aw_i, w_i)} w_i \tag{4.1}$$

Then, the iterations of the conjugate gradient algorithm can start from the optimal starting point x_p. But the application of the standard conjugate gradient algorithm does not make sure that the new direction vectors are conjugated to the vectors w_i, $i = 1, \ldots, p$. To enforce these conjugacy relations, the new direction vector d_j at iteration number j must be reconjugated to the vectors w_i through the following procedure :

$$d_j = Mg_j - \frac{(Mg_j, Ad_{j-1})}{(Ad_{j-1}, d_{j-1})} d_{j-1} - \sum_{i=1}^{p} \frac{(Mg_j, Aw_i)}{(Aw_i, w_i)} w_i \tag{4.2}$$

where M is the preconditioner of the matrix A, and g_j is the gradient vector at iteration j, $g_j = Ax_j - b$.

This is equivalent to performing the new iterations of the conjugate gradient algorithm in the subspace A-conjugate to $\text{Span}\{w_1, w_2, \ldots, w_p\}$. Then the algorithm is optimal in that sense that the actual dimension of the problem to be solved by the conjugate gradient algorithm is now equal to the dimension of A minus p.

In practice, the set of conjugate vectors (w_i) is built by accumulating the direction vectors computed for the solution of all coarse grid problems. In order to make the method more robust, in the face of finite precision arithmetic, a complete reconjugation procedure is performed :

$$d_j = Mg_j - \sum_{k=0}^{j-1} \frac{(Mg_j, Ad_k)}{(Ad_k, d_k)} d_k - \sum_{i=1}^{p} \frac{(Mg_j, Aw_i)}{(Aw_i, w_i)} w_i \tag{4.3}$$

This acceleration procedure is very efficient: for 20 V-cycles of the test problem introduced in section 2, it reduces the global number of conjugate gradient iterations for the dual Schur complement method at the coarse grid level from 450 to 150.

Of course, this acceleration technique could be applied to any problem with multiple right hand sides solved by the conjugate gradient algorithm. But for sparse linear systems of equations, the procedure should be too expensive in computing time and memory requirement. Within a domain decomposition method, the procedure is relatively inexpensive, as the reconjugation procedure involves only interface unknows, when each iteration for the condensed interface problem requires the solution of all local problems. This is all the less expensive relatively when the domain decomposition method is used only for solving the coarsest grid problem within a multigrid approach.

Table 3 details the elapsed times in the various sections of the multigrid procedure whith the solution of the coarse grid problem by the Gauss-Seidel algorithm or the accelerated dual Schur complement method. The test problem is the same as in section 2, the total number of V-cycles is 20. The global error indicated in the last column is the ratio of the L_2 norm of the error over the L_2 norm of the exact solution on the complete domain.

Table 3 : Comparison of elapsed times

coarse grid solver	computation	communication	total	global error
Gauss-Seidel	39	91	140	$7 * 10^{-4}$
dual Schur complement	35	40	75	$8 * 10^{-5}$

This table shows that the multigrid method with the accelerated Schur complement method at the coarse grid level should be faster, even on a sequential machine, than the multigrid method with Gauss-Seidel iterations at the coarse grid level. Furthermore, the first method is clearly better suited than the second one for parallel computing.

5. Numerical efficiency of multigrid with solution of the coarse grid problem by the dual Schur complement method

Table 3 exhibits different values for the global error of the solution of the problem after 20 V-cycles, depending upon the solution method for the coarse

grid problem. For both cases, the number of smoothing iterations at the fine grid level, and the stopping criterion for the residual of the coarse grid problem are the same. However, for all the test cases we studied, the results were identical: with the same stopping criterion at the coarse grid level, the multigrid algorithm converges faster when the coarse grid problems are solved by the dual Schur complement method.

A tentative explanation of this phenomenon relies upon the spectral properties of the dual Schur complement operator [**2**]. The largest eigenvalues of the dual Schur complement operator are related to the low frequencies of the primal problem. This implies that the components that are captured first by the conjugate gradient iterations for problem 3.2 are the low frequencies of the coarse grid. That is exactly what the coarse grid solver is supposed to do in order to make the multigrid method efficient.

Furthermore, these spectral properties of the dual Schur complement method allow a very fast convergence if the stopping criterion is not too small. Within the multigrid solver, the coarse grid problem needs to be solved only to within a relative residual slightly smaller than the global residual at the current cycle. Therefore, the dual Schur complement method turns out to be a very well suited iterative solver for the coarse grid level.

6. Conclusions

The results presented here show that the dual Schur complement method for solving the coarse grid problems in a multigrid algorithm has very good features from both numerical and parallel implementation points of view. Of course, although the test problem was a two-grid method, it can be used with any multigrid strategy. The acceleration technique introduced in section 4 can also be easily extended to non symmetric problems with another Krylov space method like the generalized conjugate residual. So, the coupling between multigrid and domain decomposition techniques looks very promising for the parallel solution of large scale numerical engineering problems.

References

1. C. Farhat, F.-X. Roux, *An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems*, Siam J. Sci. Stat. Comput., vol 13, pp. 379–396, 1992.
2. F.-X. Roux, *Dual and spectral properties of Schur and saddle point domain decomposition methods* , T. Chan, D. Keyes, G. Meurant, J. Scroggs, R. Voigt eds., Proceedings of the 5th International Symposium on Domain Decomposition Methods, SIAM, Philadelphia (1992), pp. 73–90 .

ONERA, Parallel Computing Division, 29 Avenue de la Division Leclerc, 92320 CHATILLON, FRANCE
E-mail address: roux@onera.fr