# Some Investigations of Domain Decomposition Techniques in Parallel CFD

F. Chalot[1], G. Chevalier[2], Q.V. Dinh[1], and L. Giraud[2]

[1] Dassault Aviation - 78 Quai M. Dassault - 92 214 St. Cloud
[2] CERFACS - 42 Av. Coriolis - 31057 Toulouse

**Abstract.** Domain decomposition methods in finite element applied aerodynamics provides a real speed-up of the convergence and good parallel scalability, even with the minimum overlap approach used here. Furthermore, a new variant of Restricted Additive Schwarz procedure is tested and shows a very attractive scalability property.

## 1 Introduction

With the emergence of the parallel distributed memory computers, certain techniques of convergence acceleration, expensive in memory, present a renewed interest for the industrial applications. Exploitation of these techniques on this type of architecture led to the use of domain decomposition decomposition techniques. This article presents the results obtained on a code of aerodynamics where we successively consider:

1. Convergence acceleration on uni-processor with a basic preconditioner (i.e. ILU(0)).
2. The analyze of the behavior of the iteration process in parallel with a domain decomposition method while trying to minimize the communication between processor with a minimum overlap approach.
3. The study of the scalability properties of the resulting code, and the test of a new approach of the additive Schwarz procedure.

We will conclude with a summary of the results obtained and a presentation of the different research path highlighted during that study.

## 2 Numerical Methods

### 2.1 Navier Stokes Equations, Discretisation

The presentation of the equations will be brief here, because no modification of the physics was made compared to the former publications related to this code [4].

The compressible Navier-Stokes equations are expressed in conservative form:

$$U_{,t} + F_{i,i} = F^d_{i,i} + \mathcal{F} \quad , \tag{1}$$

P. Amestoy et al. (Eds.): Euro-Par'99, LNCS 1685, pp. 595–602, 1999.

where $U$ is the conservative variable array: $U^T = (\rho, \rho.u, \rho v, \rho w, e)$, $F_i$ the Eulerian fluxes , $F_i^d$ the viscous or diffusive fluxes and $\mathcal{F}$ a source term.

One can rewrite the previous system in its quasi-linear form:

$$U_{,t} + A_i U_{,i} = (K_{ij} U_{,j})_{,i} + \mathcal{F} \quad , \tag{2}$$

where $A_i = F_{i,U}$ the $i^{\text{th}}$ Jacobian matrix of the Euler equations and $K_{ij}$ the diffusivity matrix.

The space-time discontinuous Galerkin approach is used and allows simultaneous space and time integration formulations. The stabilization of the method is ensured by using a Galerkin Least-Square operator. Moreover a shock-capturing term is included.

Currently we use a pseudo-time iterative scheme (predictor/corrector) because we only seek for a stationary solution. Thus the time stepping procedure, which will be also called "nonlinear iterations", is established, and leads to solve at each step a linear system $\boldsymbol{A}.\boldsymbol{x} = \boldsymbol{b}$. For a more complete description of the formulation and scheme, refer to [7] and the former publications [4].

### 2.2 Linear System

The linear system to solve at each time step is large, sparse and non-symmetric. Within the framework of this study we consider a preconditioned Krylov method for its solution. Taking into account the memory architecture on the targeted machines and the coarse grain parallelism based on the 3D unstructured grid partitioning, the domain decomposition techniques appeared as the most natural way to develop effective preconditioners. For our application, we cannot afford direct methods for the subdomain problem solutions. In that context, we considered the family of methods, generally referred under the generic name of Additive Schwarz methods, for which we can easily set up variants requiring only an approximate solution of the local Dirichlet problem. Theses techniques can be considered algebraically like local/block preconditioners with an overlap between the blocks. In order to minimize the complexity of the calculation, only one element overlap at most is permitted.

### 2.3 Formulation of the Additive Schwarz Methods

In this section we briefly present the formulation of the additive Schwarz method. Readers should refer to [8] and the references therein, for a more detailed presentation of the domain decomposition techniques.

With a partition of the unstructured finite element mesh, $\mathcal{M}$ can be decomposed in $N$ disjoints sets of elements (sub-domains), or equivalently into $N$ overlapping sets of nodes. Let's call $\{W_i\}_{i=1,N}$ the subsets of nodes and $W$ the complete set of nodes over the entire grid. This drives to: $W = \bigcup_{i=1}^{N} W_i$, which can be called a "partition with a minimum overlap" of Of that element partition, one can deduce a strict node partition by allocating each interface node to a single subset of elements $W_i^0 \subset W_i$. This leads to: $W = \bigcup_{i=1}^{N} W_i^0$ with $W_i^0 \cap W_j^0 = \emptyset$ for $i \neq j$.

We can then associate with each of the $W_i^0$ a canonic restriction operator $R_i^0$. We can define in the same way the restriction operators $R_i$ for the sets $W_i$. Using the above notations the local problems is defined by:

$$A_i = R_i A R_i^T \,, \tag{3}$$

where $R_i^T$ is a prolongation operator.

These notations allows the formulation of the basic Schwarz additive preconditioner like: $M_{AS} = \sum_{i=1}^{N} R_i^T A_i^{-1} R_i$.

These notations allows to describe the preconditioner recently introduced by [2, 1] whom called it Restricted Additive Schwarz: $M_{RAS} = \sum_{i=1}^{N} (R_i^0)^T A_i^{-1} R_i$. In the case of the partition of the elements in a Finite Element formulation, the construction of the local problems as defined by the Equation (3) requires the assembly of the stiffness matrix for the interface nodes.

The exact solution of the local problems which appear with the $M_{AS}$ et $M_{RAS}$ being too costly for an industrial problem, we substituted them by approximate resolutions calculated via an ILU(0). In order to limit the communications during the assembly of the local stiffness matrices $A_i$ on the interface nodes, a new variant have been implemented, which consists in assembling only the diagonal block associated with each of the interface nodes. Thus we carried out experiments with three preconditioners:

1. $M_{ILU(0)-AS}$ : Additive Schwarz with an inexact local solution using ILU(0).
2. $M_{ILU(0)-dAS}$ : Additive Schwarz with inexact local solution using ILU(0) and only the assembly of the interface nodes diagonal block.
3. $M_{ILU(0)-dRAS}$ : Restricted Additive Schwarz with inexact local solution using ILU(0) and only the assembly of the interface nodes diagonal block.

These preconditioners have been implemented in order to accelerate the GMRES (Generalized Minimum RESidual) method introduced by Y. Saad and M. Schultz [5]. The size of the Krylov space is limited to 20 in 2D and at 10 in 3D. Moreover the tolerance $\varepsilon$ controlling the decrease of the 2-norm of the GMRES' residuals is fixed at $10^{-1}$.

## 3 Implementation & Results

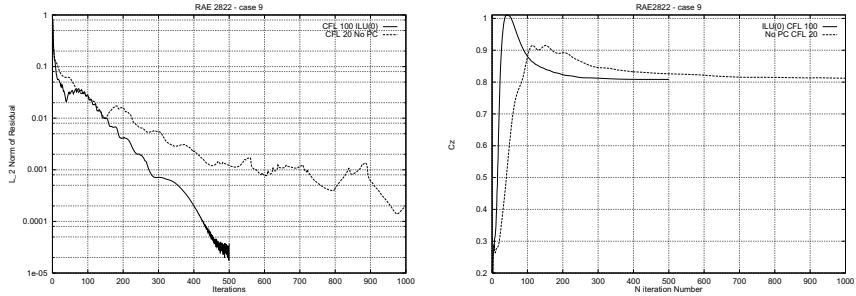
### 3.1 Sequential Linear System : ILU(0) as Preconditioner

We present here the results obtained in sequential (mono-processor) on a 2D turbulent test case representative of real flow conditions. It is RAE2822 test case number 9 (cf. [3]). The test conditions are those of a turbulent transonic flow: $R_{e_c} = 6.5\,10^6$, $\alpha = 2.79°$, $M = 0.73$.

The grid is the cut in triangles of a structured fine grid with $362 \times 64$ or 22900 points and 45000 elements. In the results presented here, we used a two-layer $k - \varepsilon$ turbulence model.

One can report the following observation on the sequential tests:

1. logically, the number of Krylov spaces (*i.e.* the number of matrix-vector products) necessary to the solution of the linear systems decreases dramatically when the ILU(0) preconditioner is applied, with a fixed tolerance $\varepsilon$ and CFL condition (*i.e.* with the same "speed" speed in the time marching procedure),




(a) Residual *vs* Newton It. (b) Lift *vs* Newton It.

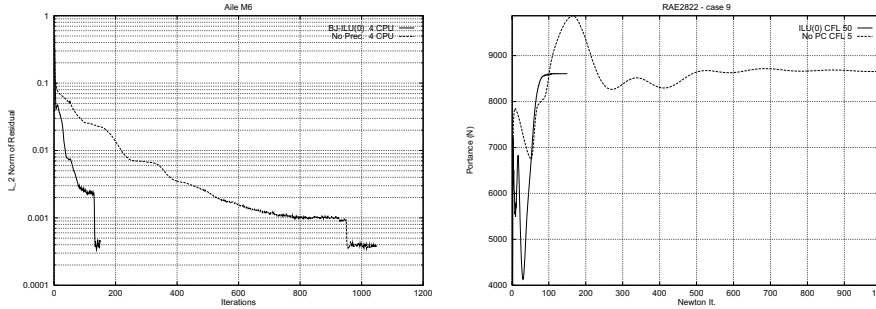
**Fig. 1.** RAE 2822 Case 9 - dashes: CFL=20 No Prec. - solid: CFL=100 ILU(0)

2. the preconditioning by itself brings little to the nonlinear process (the "outer" pseudo-time-stepping iterations) if we do not modify the CFL condition when compared to the unpreconditioned case. Indeed one can then have a gain or a lost of CPU time according to the ill-conditioning of the linear systems, but overall the evolution of the nonlinear residues is similar.
3. If, on the opposite, we take advantage of the improvement of the robustness of the preconditioned system to increase the CFL, then the nonlinear residuals converge much more quickly ($\div 2$ on the CPU time here). Figure 1(a) compare the evolution of the residuals for a unpreconditioned calculation with a maximum CFL condition of 20, and a ILU(0) left-preconditioned computation with a maximum CFL of 100.
   The same acceleration of convergence can be noted on Figure 1(b) which present the evolution of the lift coefficient during the iterations.
   Acceleration comes primarily from the increase in the CFL condition authorized by the use of the preconditioner. Eventually and unfortunately we encounter a limit of stability of the scheme and in practice the CFL lie between 50 (coarse grid) and 300. Beyond that, oscillations appear and calculation does not converge any more. Nevertheless the multiplication by 5 or 15 of the CFL is sufficient to obtain a 2 to 3 fold acceleration over the computational time.
4. For *steady calculations* the value of the tolerance $\varepsilon$ with constant CFL does not seem to have a radical effect on the global time-marching process of the nonlinear iterations: this considering the transient phase or on the level of final residue. Nevertheless the use of the preconditioner ILU(0) facilitates

the resolution of the linear system. Let us note that that should present an interest for the unsteady simulations where the value of $\varepsilon$ must be relatively low ($\leq 10^{-2}$).

### 3.2 Study of the M6 Wing Test Case on 4 to 32 Processors

**Parallel : Efficiency of the ILU(0) with $M_{ILU(0)-dAS}$** The approach known as domain decomposition is used to reduce the degradation induced in the preconditioner by the localization of the data on each processor; that localization is mandatory for effective parallelism. The procedure of additive Schwarz with minimal overlap used here constitutes an ab-initio approach of domain decomposition and can be developed without the choice and use of dedicated libraries.




(a) Residual *vs* Newton It. (b) Efforts *vs* Newton It.

**Fig. 2.** M6 wing - 4CPU - dashed: CFL=5 Non Prec. - solid: CFL=100 - $M_{ILU(0)-dAS}$

We will use thereafter the case of the ONERA M6 wing to test the parallel applications for the 3D turbulent flows. The simulation parameters are: $R_e = 11.7\,10^6$, $M = 0.844$, $\alpha = 4.56°$ [6]. The grid consists of 77000 points, divided into 4, 8, 16 and 32 subdomains. The size of problem prevented the realization of a sequential test, so the comparison was made between the calculation on 4 processors and the later decompositions. The decomposition itself have been carried out by Dassault-Aviation, taking care to balance the distribution of the points *and* elements over the domains. The stopping criterion of the GMRES is $\varepsilon = 10^{-1}$, the size of the Krylov space is 10 with and we allow only 2 more restarts of the GMRES.

Figure 2(a) compares the residuals of a non-preconditioned calculation with a CFL of 5 with those of a $M_{ILU(0)-dAS}$ calculation with a CFL of 50 (these two CFL are roughly the acceptable maximum for each calculations). We can note that the lift history (Figure 2(b)), shows clearly the acceleration in the establishment of the flow, as do the convergence history.

Finally, we can note that the CPU time has been decreased by a factor of 3 with the use of $M_{ILU(0)-dAS}$: from a total of 7500 units of time for the un-

preconditioned case we ended to 2000 units of time with the preconditioner, both of the calculations have been performed on 4 CPU.

**Parallel : Scalability of the $M_{ILU(0)-dAS}$** Globally, there is an independency of the residual evolution during nonlinear iterations with respect to the number of processors. Furthermore the CPU time gathered in the Table 1 reveal a good efficiency, in spite of a minimum overlap and a basic preconditioner. The efficiency is calculated by taking as reference time the calculation on 4 processors.

**Table 1.** M6 wing - CFL=50 - $M_{ILU(0)-dAS}$

| | 4 Domains | 8 Domains | 16 Domains |
|---|---|---|---|
| CPU Equiv. Time | 1922 u | 1031 u | 558 u |
| Acceleration | – 1 – | 1.86 | 3.44 |
| Efficiency | – 1 – | 0.93 | 0.86 |
| # of Krylovs | 1259 | 1423 | 1493 |

This is not a sufficient indication to characterize the scalability of the preconditioner; Table 1 also presents the total number of internal GMRES iterations needed to get the appropriate convergence.

We can see that, according to the total number of Krylov spaces, the quality of the local preconditioner degrades with the number of subdomains. The relatively good scalability in CPU times is partly due to a better memory access resulting of the reduction of the problem's size (mainly cache effects).

These results are nevertheless encouraging; they prove that an domain decomposition approach with a minimal overlap strategy presents an sufficient parallel efficiency for applications up to 16 and probably 32 processors.

**Effects of Periodic Preconditioner Estimation** We have constated that it is no necessary to recompute the preconditioner for each nonlinear iteration, we use the an "old" evaluation of the preconditionner for the current iteration.

For example on the 4 domains M6 wing test case, the quality of the preconditioner suffers when we compute it only every five Newton iterations (1409 Krylovs *vs.* 1259). But that degradation is largely compensated by the CPU savings of the non-evaluation of the ILDU(0): In our case the gain is clear: on 4 CPU with $M_{ILU(0)-dAS}$ it drives the CPU time from 1922 units to 1626.

That strategy was verified for all the domain decomposition techniques considered here, and on 4, 8, 16 or 32 processors.

**Comparison of $M_{ILU(0)-dAS}$ and $M_{ILU(0)-AS}$** The $M_{ILU(0)-dAS}$ preconditioner quality can be improved if we not only assemble the diagonal block, but all the blocks of the line in the matrix for each interface node. These additional assemblies require the enrichment of the information that describes the

interface (additional handling of the grid) as well as a notable increase in the communications to build the local systems.

The use of $M_{ILU(0)-AS}$ allows a gain of approximately 15% of the number of matrix-vector products for each nonlinear iteration. Moreover, the additional information to exchange increases the size of the message but not the number of them. Thus on hight-bandwidth machines, the overcost of communication related to a higher volume of data exchanged can be neglectible.

The saving of time CPU is thus 10 to 15% compared to $M_{ILU(0)-dAS}$. Nevertheless, the complexity of the connectivities sorting procedure, the non-control of the user on the interface region, makes its use cumbersome without the use of a domain decomposition library dealing with these aspects at the mesh partitioning stage.

### 3.3 Restricted Additive Schwarz

We use here the Restricted form of the additive Schwarz procedure as introduced in the section 2.3, wich differs only at the preconditioning step in the GMRES: we no more assemble the result of the preconditioning, but rather affect the value of the node's domain to the others.

**$M_{ILU(0)-dRAS}$ Scalability** We consider a similar study as the one described in Section 3.2 but for the $M_{ILU(0)-dAS}$ with the restricted additive Schwarz procedure and a periodic evaluation of the ILU(0) every other five Newtons iterations. As previously reported, the residual evolution against the nonlinear iteration is not influenced by the successive division into 4, 8, 16 or 32 subdomains. On the other hand, and in a surprising way, the internal iteration count of the GMRES *decreases* when one goes from 4 to 8 then 16 fields (cf Table 2). This is not observed any more on the switch from 16 to 32 processors.

**Table 2.** M6 wing - CFL=50 - $M_{ILU(0)-dRAS}$ - Periodicity : 5

| | 4 Domains | 8 Domains | 16 Domains | 32 Domains |
|---|---|---|---|---|
| CPU equiv. Time | 1583 u | 795 u | 398 u | 225 |
| Acceleration | – 1 – | 1.99 | 3.97 | 7.02 |
| Efficiency | – 1 – | 0.995 | 0.993 | 0.878 |
| # of Krylovs | 1244 | 1202 | 975 | 1222 |

The CPU times gathered in Table 2 show a remarkable scalability of the method.

## 4 Conclusion & Future Work

During this study, it was possible to adapt a semi-industrial aerodynamic code in order to test quickly some domain decomposition ideas and preconditioners for

practical simulation using coarse grained parallelization on distributed memory architectures.

The choices made during this study were directed towards a minimization of the number of exchanges between processors. Communications are only need by the matrix-vector products, preconditioning, scalar products and for the diagonal block assembly of the Jacobian matrix on the interface nodes. Moreover the ILU(0) was choosed with the same simplicity/robustness requirements. Despite all its intrinsic limitations compared to more traditional approaches of domain decomposition, that technique showed its attracting potential on various aerodynamics simulation test cases. In this study, a 2 to 3 fold acceleration has been observed when compared to the best unpreconditioned implicit calculations, and the very good scalability of the $M_{ILU(0)-dRAS}$ has been demonstrated.

Most of the acceleration gain is a consequence of the increase in the robustness which allowed the increase in the CFL. Depending on the flows/grids tested, the CFL limit lies between 50 and 300. One can question whether further increase the CFL limits would still be so beneficial? And, how to achieve these CFL conditions (integration schemes, linearization schemes, boundary conditions).

## References

[1] X.-C. Cai, C. Farhat, and M. Sarkis. A minimum overlap restricted additive Schwarz preconditioner and applications in 3D flow simulations. In *Proceedings of the 10th Intl. Conf. on Domain Decomposition Methods*, pages 479–485, Providence, 1998. AMS.

[2] X.-C. Cai and M. Sarkis. A restricted additive schwarz preconditioner for general sparse linear systems. Technical Report Tech Report CU-CS-843-97, Dept. of Comp. Sci., Univ. of Colorado at Boulder, 1997. to appear in SIAM SIAM J. Sci. Comput.

[3] P.H. Cook, MA McDonald, and MCP Firmin. Aerofoil RAE 2822 - pressure distributions and boundary layer and wake measurements. In *AGARD AR 138*, pages A6, 1–77, 1979.

[4] Thomas J.R. Hughes, Leopoldo P. Franca, and Michel Mallet. A new finite element formulation for computational fluid dynamics: VI. convergence analysis of the generalized SUPG formulation for linear time-dependant multidimentional advective-diffusive systems. *Comp. Meth. in Applied Mechanics and Engineering*, (63):97–112, 1987.

[5] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, 1986.

[6] V. Schmitt and F. Charpin. Pressure distribution on the ONERA M6 wing at transonic mach numbers. In *AGARD R-702*, pages B1, 1–44, 1982.

[7] Farzin Shakib, Thomas J.R. Hughes, and Zdeněk Johan. A new finite element formulation for computational fluid dynamics: X. the compressible euler and navier-stokes equations. *Comp. Meth. in Applied Mechanics and Engineering*, (89):141–219, 1991.

[8] B.F. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, New York, 1st edition, 1996.