Contemporary Mathematics
Volume 157, 1994

A Domain Decomposition Environment for Local Time Dependent Problems

F. DELLAGIACOMA, S. PAOLETTI, F. POGGI, M. VITALETTI

ABSTRACT. A framework for parallel computations of local time dependent problems is introduced. It is based on *overlapping domain decomposition* techniques. We also present two examples of gas dynamics simulations. Numerical results show how the system can be useful in dealing with a variety of applications.

1. Introduction

The evolution in time of many physical phenomena is described mathematically by partial differential equations (PDE), whose solution is the primary goal of numerical algorithms. Initial boundary value problems are encountered in several application areas, ranging from the air flow around an airplane to the heat diffusion between bodies at different temperatures, just to mention a few of them. For this class of problems there are by now well established methods (e.g. finite differences, finite elements, spectral techniques), which are able to describe the observed behavior. Three main challenges are represented (1) by the complex geometries one needs to describe in practical applications, (2) by the long elapsed times involved in numerical simulations, and (3) by the high computational costs typically required by accurate schemes. It is therefore natural to consider the approach based on *domain decomposition* as a good candidate to construct appropriate solutions. This is because the partition of the whole three dimensional region of space into individual subdomains leads to computational units of smaller size, which are to a large extent independent of each other. As a result the computation can be performed in parallel, thus decreasing the overall turn around times, provided that the communication/computation ratio is a favorable one.

1991 Mathematics Subject Classification: 65M55, 76H05, 76N15.
This paper is in final form and no version of it will be submitted for publication elsewhere.

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

In this paper we introduce a framework, called Paragrid, suitable for the multidomain treatment of time dependent, spatially localized problems, described in general by non-linear systems of PDE. The major underlying assumption concerns the grids associated with each block, which need to be *structured* and are required to *match* at the internal boundaries. Although the time evolution of the relevant variables is recorded through single time levels, multistep algorithms (e.g. Runge-Kutta) can also be applied. No further restriction is imposed on the numerical algorithm. Therefore explicit as well as implicit schemes can be adopted according to the specific situation one is interested in. In addition to the class of problems mentioned above, the conceptually more difficult case represented by non-local *elliptic boundary value* problems might be tackled within Paragrid, by seeking for the steady-state solution of the corresponding fictitious initial boundary value equations. Another way of obtaining the solution of the original problem is by resorting to an iterative procedure.

The structure of this paper is as follows. In section 2 we describe the building blocks of Paragrid, focusing the attention on its general features, instead of pointing out specific details depending upon any particular implementation.

The study of the inviscid compressible flow for two selected geometries (i.e. around a sphere and around a swept wing) is the subject of section 3. The emphasis here is on the effects of the multiblock procedure on the convergence of the iterative algorithm and on the parallel performance obtained using a distributed memory system.

2. Paragrid

In the past most effort in the area of domain decomposition has involved the theoretical study and development of numerical algorithms for elliptic PDE. In the literature two main approaches are followed: the one based on *overlapping* domains, which traces to the pioneering work of Schwarz and the *nonoverlapping* method, whose basic idea amounts to find the *reduced interface operator* at the internal boundaries. Recently domain decomposition approaches have also been applied to time dependent problems, like for example, those arising in computational fluid dynamics[1] and being described by either hyperbolic[2], or parabolic equations, or even by systems of the mixed type (e.g. elliptic-hyperbolic). The non linearity of the equations represents a major complication, which makes it difficult to design efficient numerical algorithms applicable to cases of practical relevance. It is for *time dependent* problems with the further property of being *local* in space that we introduce Paragrid: a parallel multidomain environment.

The partition of the spatial domain into adjoining, nonintersecting subdomains must be built beforehand. The outcome of the grid generation process results in a set of discretized grids, for which the further requirement of being structured and boundary-fitted has been imposed. The first action of Paragrid is to perform a thorough analysis of each individual block in order to find all internal boundaries shared by adjacent subdomains. As the ultimate goal of the numerical algorithm is to determine the evolution in time of field variables, whose value in a given point depends upon the values in a set of neighboring

points, some care must be taken at the internal interfaces and in their vicinity. In fact for such grid points the knowledge of the field in the neighboring domain might be necessary. This requirement is met by extending the originally nonoverlapping subdomains (i.e. the *cores*) in such a way as to include a certain number of additional points, thus forming a so-called *interaction region*. The size of the interaction region is controlled by a geometric parameter, which is specified in input. As a result of the enlargment procedure, data belonging to the overlap zones are duplicated to insure consistency of the computation carried out in each individual domain. The exchange of the relevant information at each time step brings up the important issue of the updating cycles concerning dynamic variables. As a matter of fact three stages characterize every single cycle. Namely:

- the data **IMPORT** from core volumes into the overlap regions. Such data are known at the current time instant.
- the field variables **UPDATE** performed by the chosen numerical algorithm.
- the **EXPORT** of the updated quantities from cores into interaction zones, where these quantities will be used in the next updating cycle.

After each time step the partial solutions are brought together to form the global solution. From the structure of the time marching strategy illustrated above it emerges that computations relative to individual subdomains can be performed in parallel, data exchange along with synchronization being required only at the end of each updating step. As the framework has been designed to take care of many tasks including process scheduling, communication and even dynamic load balancing, it turns out to be a powerful tool to study a broad class of algorithms and their application to complex geometries. Moreover the coordination of the single computations is achieved through general procedures, which are independent of both the physical nature of the field variables and of the parallel computer architectures. It is important to emphasize that the interaction between adjacent subdomains is managed through initial conditions expressed in the overlap regions. Although one step explicit formulas for time integration fit naturally in this framework, multistep explicit methods can be incorporated without much effort, as the system allows the user to keep track of the variables as computed in previous stages. Special attention deserve implicit algorithms, because the new field values depend on points of the whole space at the previous time. Since only local field equations can be managed within Paragrid, the assumption of admitting exchange of information between pairs of neighboring domains, disregarding any interaction with those lying further away, must be carefully justified.

3. Numerical Results

In the last section we have outlined the basic overlapping domain decomposition strategy of Paragrid. Let us now study the external compressible flow

in three dimensional space, surrounding two objects, whose geometry has been chosen for illustrative purposes. For simplicity let us focus our attention on the inviscid case, which is described by the Euler equations. Using the strong conservation law they are expressed as follows:

$$\partial_t \mathbf{q} + \partial_\xi \mathbf{F} + \partial_\eta \mathbf{G} + \partial_\zeta \mathbf{K} = 0 \tag{1}$$

where $\mathbf{q}$ denotes the array of the dynamic variables containing mass, momentum and energy densities, $\mathbf{F}$, $\mathbf{G}$, $\mathbf{K}$ are the flux vectors and the body-fitted curvilinear coordinates are denoted as $\xi = \xi(x, y, z, t)$, $\eta = \eta(x, y, z, t)$, $\zeta = \zeta(x, y, z, t)$.

Time integration is performed through an implicit Euler backward algorithm of the first order, whereas space derivatives are discretized by a centered finite difference scheme. Furthermore the flux vectors are expanded in terms of the discrete variables $\mathbf{q}^n = \mathbf{q}(n\Delta t)$. Keeping only linear terms and applying the Alternating Direction Implicit (ADI) approximation[3] to the differential operator, we have to solve the linear system of algebraic equations:

$$\begin{aligned}[\mathbf{1} + \Delta t \partial_\xi \mathbf{A}^n]\,[\mathbf{1} + \Delta t \partial_\eta \mathbf{B}^n]\,[\mathbf{1} + \Delta t \partial_\zeta \mathbf{C}^n]\,\Delta \mathbf{q}^n = \\ -\Delta t\,[(\partial_\xi \mathbf{F}^n) + (\partial_\eta \mathbf{G}^n) + (\partial_\zeta \mathbf{K}^n)]\end{aligned} \tag{2}$$

with $\mathbf{A}^n = \partial \mathbf{F}^n / \partial \mathbf{q}$, $\mathbf{B}^n = \partial \mathbf{G}^n / \partial \mathbf{q}$, $\mathbf{C}^n = \partial \mathbf{K}^n / \partial \mathbf{q}$ being the Jacobians of the flux vectors and $\Delta \mathbf{q}^n = \mathbf{q}^{n+1} - \mathbf{q}^n$. A particular form of the artificial dissipation[4] is added to eliminate from the solution spurious oscillations and to enable the use of large values of the Courant number.

The system is solved for a medium size (40,375 nodes, 180,480 equations per time step), three dimensional problem corresponding to the compressible transonic flow around an ONERA M6 swept wing. The numerical results illustrating the convergence history of the algorithm are shown in Figure 1 and Figure 2.

Two different decompositions (i.e. 15 and 30 blocks) are compared with the computation performed on a single structured grid covering the entire domain. It is seen that doubling the number of domains does not penalize too seriously the number of iterations necessary to reach the stationary state. Furthermore, by increasing the size of the overlap (i.e. the number of grid layers) between adjacent blocks, convergence is achieved much faster, even though the computational cost per domain is higher and a larger amount of data needs to be exchanged at each time step.

One of the major advantages of domain decomposition is represented by its potential for parallelization. We have studied such aspect in the framework of Paragrid by running applications on a distributed memory system formed by several workstation. As an example we report in Table 1 results obtained for the simulation of the external compressible flow around a sphere. Even by connecting just a few computational nodes, remarkable performances compared to the sequential runs are obtained . A key role in parallel implementations is played by load balancing, since for homogeneous networks the total elapsed time is basically determined just by the most loaded processor. Another crucial aspect concerns high speed communication, which might further improve the overall

Figure 1: Convergence history of inviscid transonic flow for a given angle of attack α , Mach number M_∞ and overlap = 1.

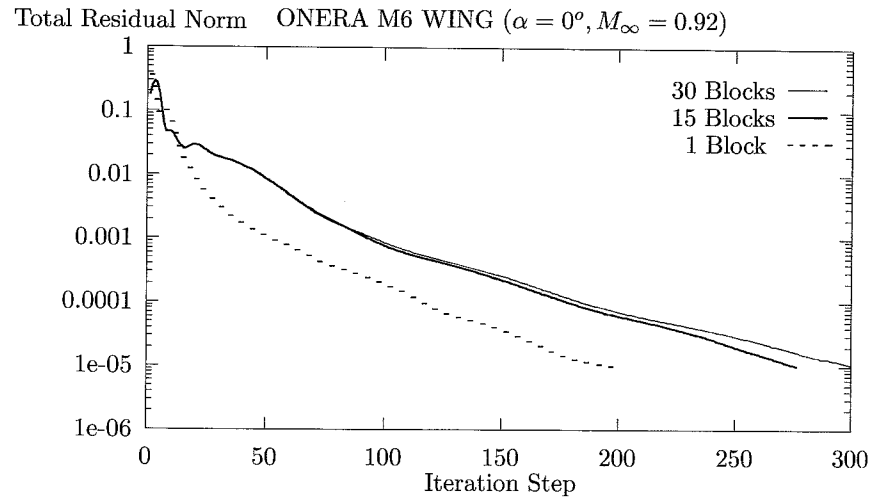
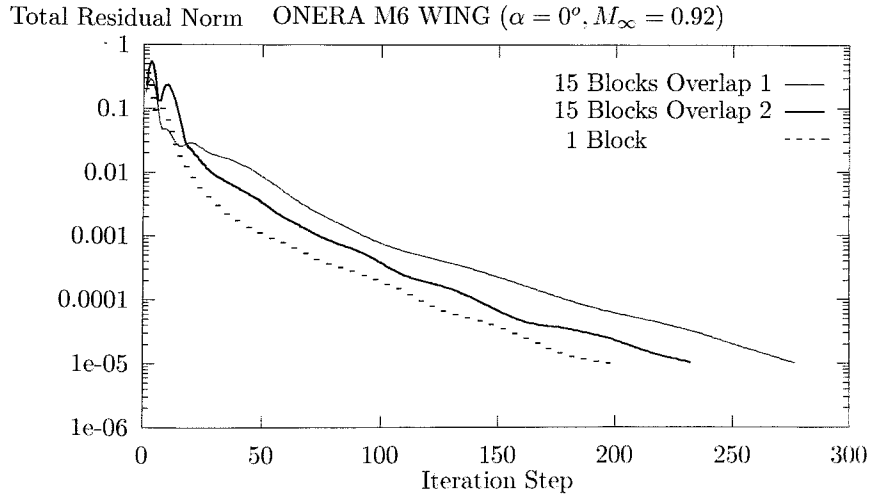


Figure 2: Effect of the overlap in the multiblock calculation of the convergence history of the same inviscid transonic flow as in Figure 1.



performance, especially for those applications for which data exchange happens to occur very often.

From our analysis it should be evident that the attractive features of domain decomposition ideas can be combined with the advantages offered by parallel architectures. This opens up the possibility of studying realistic applications involving complex geometries, but exploiting advanced numerical techniques.

Table 1: Compressible flow simulation around a sphere.
Performance vs. N (number of IBM RS/6000 mod. 530H workstations)

N	Elapsed Time [sec]	Speed Up
1	6698	1
3†	2268	2.95
4‡	1845	3.63

† Nodes connected via SOCC (Serial Optical Channel Converter).
‡ Nodes connected via Token Ring.

References

1. T.F. Chan, *Domain Decomposition Algorithms and Computational Fluid Dynamics*, in "Vector and Parallel Computing Issues in Applied Research and Development", J.Dongarra, I. Duff, P. Gaffney and S. McKee (Editors), E. Horwood Ltd., Chichester, UK, 1989, pp. 65-82.

2. A. Quarteroni, *Domain Decomposition Methods for Systems of Conservation Laws: Spectral Collocation Approximations*, SIAM J.Sci.Stat.Comput., Vol. II, No. 6 (1990), 1029-1052.

3. R.M. Beam and R.F. Warming, *An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form*, J.Comp.Phys., Vol. 22, No. 1 (1976), 87-110.

4. S. Paoletti, M. Vitaletti and P. Stow, *An Unfactored Implicit Scheme for 3D Inviscid Transonic Flows*, 10th AIAA Applied Aerodynamics Conference, Palo Alto, CA, 1992, pp. 518-527.

IBM Semea
European Center for Scientific and Engineering Computing
Viale Oceano Pacifico 171/173, 00144 Rome, Italy
E-mail: dellagia@romesc.vnet.ibm.com