65

Multidomain Finite Elements and Finite Volumes for Advection-Diffusion Equations

M.-C. Ciccoli, R. L. Trotta

1 Introduction

New interface conditions are proposed for domain decomposition methods in the advectively dominated limit of advection-diffusion problems.

Domain decomposition methods are interesting for several reasons. They allow a simplification of the geometry, a reduction of the size of the problems, and the use of different physical and/or numerical models on the different subdomains in order to get a more accurate modelization of the flow. Furthermore DD methods are easily parallelizable.

Unfortunately, DD algorithms that work well for viscosity dominated flows can perform very badly for convectively dominated flows. This is due to the fact that the matching conditions, although mathematically correct, may not respect the hyperbolic limit of the advection-diffusion equation.

The adaptive DD algorithms enforce the appropriate interface condition in respect with the "direction of the wind". We are interested in the capability of these algorithms to efficiently solve an advection-diffusion equation in view to apply them to the Navier-Stokes equations.

Experiments in this article have been made using two different discretization methods: finite element and finite volume/finite element.

We investigate the properties of the Adaptive Dirichlet Neumann (ADN) and the Adaptive Robin Neumann (ARN) algorithms, proposed by Carlenzoli and Quarteroni [CQ95]. We also study the performances of two new algorithms denoted d-ADN and d-ARN which are *damped* versions of the ADN and ARN algorithms and have been constructed to improve the convergence of the ADN and ARN algorithms. We will see that the d-ADN and d-ARN algorithms do not get exactly the right solution, but the error they introduce is acceptable for small diffusion coefficient which is the case we are interested in and they converge significantly faster than the ADN and ARN algorithms.

Ninth International Conference on Domain Decomposition Methods
Editor Petter E. Bjørstad, Magne S. Espedal and David E. Keyes

Regarding other recent works on domain decomposition methods for advection-diffusion problems, we mention [NR95], and [TB95]. These two works are made in the framework of finite difference discretizations. In [NR95] A Schur type formulation with outflow boundary conditions using overlapping subdomains decomposition is constructed, while in [TB95] a way to improve the Schwarz algorithm convergence for advection-dominated cases is proposed.

2 The Advection-Diffusion Boundary Value Problem

Let Ω be a bounded, connected, open subset of $I\!R^2$ with a Lipschitz continuous boundary $\partial\Omega$ and denote by $\mathbf{n}$ the unit outward normal direction on $\partial\Omega$. Let $\mathbf{b} = \mathbf{b}(\mathbf{x})$ denote the given flow velocity, $\varepsilon = \text{const} > 0$ denote the diffusivity, and $a \geq 0$ the absorption coefficient. Let $\{\partial\Omega^{in}, \partial\Omega^{out}\}$ be a partition of $\partial\Omega$, where

$$\partial\Omega^{in} = \{\mathbf{x} \in \partial\Omega \mid \mathbf{b}\cdot\mathbf{n} < 0\} \qquad \textit{inflow boundary}$$

$$\partial\Omega^{out} = \partial\Omega \setminus \partial\Omega^{in} \qquad \textit{outflow boundary}.$$

The *scalar steady advection-diffusion boundary value problem* consists of finding $u = u(\mathbf{x})\ \forall \mathbf{x} \in \bar{\Omega}$ such that

$$(AD)\ \begin{cases} L_\varepsilon u := -\varepsilon\Delta u + div(\mathbf{b}u) + au = f & \text{in } \Omega \\ u = g & \text{on } \partial\Omega\,, \end{cases} \tag{2.1}$$

where the given body source $f : \Omega \to \mathbf{R}$ and $g : \partial\Omega \to \mathbf{R}$ are prescribed data.

In the advection-dominated regime, i.e. when $\dfrac{\varepsilon}{\|\mathbf{b}\|} \ll 1$ the solution u will vary rapidly in layers of width $O(\varepsilon)$ at the outflow boundary $\partial\Omega^{out}$ and in layers of width $O(\sqrt{\varepsilon})$ across characteristic curves issuing from discontinuous boundary data.

If layers arise, the numerical method may produce oscillatory solution, if this situation is not properly faced, resorting either to a stabilization method, like SUPG, GALS, DW (see [BBF+92]) or to an upwind formulation.

DD methods are interesting for such problems because, beyond the classical reasons, the problem can be stabilized only in the subregions where it is necessary.

3 Adaptive DD Methods

To simplify our presentation, we consider a partition of the initial domain Ω into two nonoverlapping subdomains, Ω_1 and Ω_2. The generalization to several subdomains is straightforward. Let Γ be the interface between Ω_1 and Ω_2, $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$. Let u_i be the restriction of u on Ω_i and $\mathbf{n}_i$ the outward normal unit vector on Γ.

The multidomain formulation can be obtained by decoupling the resolution in Ω_1 from the resolution in Ω_2, and by assigning properly matching conditions on the interface Γ. Natural transmission conditions arise from the requirement that u is the solution to the overall problem (hence, in particular, u is sought in a precise functional space), and that u_k must be the restriction of u to Ω_k, $k = 1, 2$.

For elliptic problems these conditions are the continuity of the solution and the continuity of the flux across the interface Γ.

The well known Dirichlet/Neumann algorithm (see [BW86] and [MQ89]) assigns, at each iteration of the procedure, a Dirichlet condition to one subdomain and a Neumann condition to the other one.

It works well for viscous flows, but for advection dominated problems can produce at each iteration nonphysical layers because the matching condition may not respect the hyperbolic limit of the advection-diffusion equation. The Dirichlet condition at the outflow, prescribing specific values, can generate artificial layers.

This idea is the basis of the adaptive DD methods proposed by Carlenzoli and Quarteroni in 1993. The first method proposed is the ADN method; in this algorithm, a Dirichlet condition is imposed at the part of the interface (on Γ_i^{in}) where the "flow is coming in" the domain, inversely a Neumann condition is used where (on Γ_i^{out}) the "flow is going out of" the domain. The Adaptive Dirichlet/Neumann algorithm starts with (u_1^0, u_2^0) and constructs the iterates (u_1^k, u_2^k) by solving the following problems:

$$\text{(ADN)} \qquad \begin{cases} L_\varepsilon u_1^k = f & \text{in } \Omega_1 \\ u_1^k = \theta u_2^{k-1} + (1-\theta) u_1^{k-1} & \text{on } \Gamma_1^{in} \\ \dfrac{\partial u_1^k}{\partial n_1} = -\dfrac{\partial u_2^{k-1}}{\partial n_2} & \text{on } \Gamma_1^{out} \end{cases} \qquad \begin{cases} L_\varepsilon u_2^k = f & \text{in } \Omega_2 \\ u_2^k = \theta u_1^k + (1-\theta) u_2^{k-1} & \text{on } \Gamma_2^{in} \\ \dfrac{\partial u_2^k}{\partial n_2} = -\dfrac{\partial u_1^k}{\partial n_1} & \text{on } \Gamma_2^{out} \end{cases} \tag{3.2}$$

θ is a relaxation parameter introduced to improve the convergence of the algorithm.

The other adaptive domain decomposition method, ARN, arises from the fact that for advection dominated problems the continuity has to be weakly enforced [Qua90]. Thus, in the ARN algorithm, the Dirichlet condition is replaced by a Robin condition, that means we impose:

$$\varepsilon \frac{\partial u_1}{\partial \mathbf{n}_1} - \mathbf{b} \cdot \mathbf{n}_1 u_1 = -\varepsilon \frac{\partial u_2}{\partial \mathbf{n}_2} + \mathbf{b} \cdot \mathbf{n}_2 u_2 \qquad \text{instead of} \qquad u_1 = u_2 \qquad \text{on } \Gamma_i^{in}$$

The two approaches, ADN and ARN, are equivalent if $\varepsilon \to 0$ and $\Gamma_i^0 = \{\mathbf{x} \in \Gamma \mid \mathbf{b} \cdot \mathbf{n}_i = 0\}$ is negligible. For more details on the two algorithms, see [Tro96], [Cic].

The Damped Versions

It is well known that for pure hyperbolic equations (i.e. $\varepsilon = 0$) the boundary conditions have to be given only on the inflow part of the boundary. As far as domain decomposition methods are concerned, one has to give also the transmission conditions, but only on the inflow part of the interface Γ. But, even in the case $\varepsilon = 0$, the DD algorithms enforce a derivative continuity at the outflow part of the interface. This slows down the convergence rate of the algorithms.

This remark leads to the construction of two new algorithms denoted d-ADN (damped ADN) and d-ARN (damped ARN) by changing the Neumann condition of the normal derivative continuity into the hyperbolic outflow condition.

Of course, the damped algorithms do not provide exactly the right solution for $\varepsilon \neq 0$. In particular, the error between two d-ARN solutions at the interface satisfies the inequality

$$\int_\Gamma \left(b.n_1 - \frac{\varepsilon}{2}\right)(u_1 - u_2)^2 \leq \frac{\varepsilon}{2}\int_\Gamma \left(\frac{\partial u_2}{\partial n_2}\right)^2$$

This error decreases proportionally to ε and is, obviously, equal to zero for $\varepsilon = 0$.

We are satisfied if the order of the error introduced by the damped formulation is the same than this of the discretization error.

We refer to [Tro96] and [Cic] for a more complete analysis of the errors of the damped methods.

The Discretized DD Algorithms

We do not give extended explanations for the discretizations of the DD algorithms. See [Tro96] for the finite element discretization and to [Cic] for the finite volume/finite element discretization. In the finite element approximation we discretize the advection-diffusion equation using a P1 Galerkin method. In the mixed finite volume/finite element formulation, we use an upwind finite-volume approximation for the convective term and a finite-element approximation using the P_1 basis function for the diffusive term.

4 Validation on Test Cases

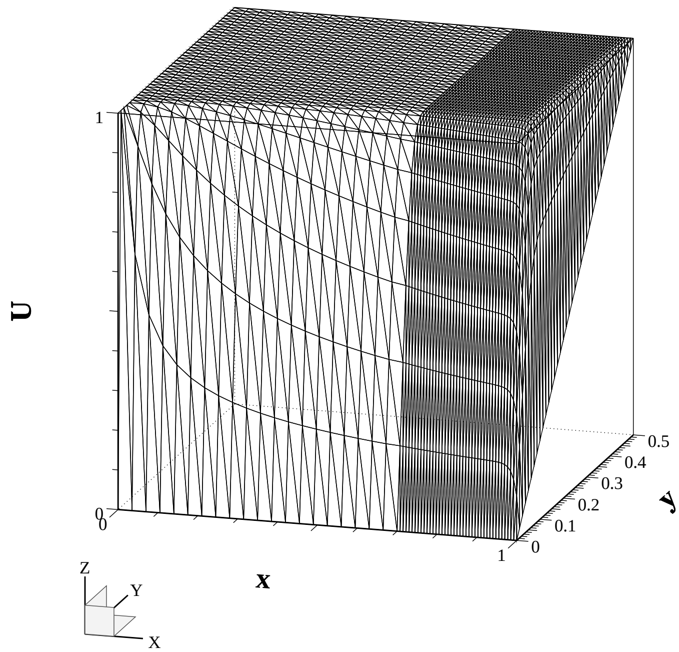
The schemes are implemented on a cluster of an IBM RS/6000 (model 560 and 950) workstations connected by Ethernet, as well as on IBM-9706-SP1 parallel distributed memory machine. We have used both the finite element and finite volume/finite element discretizations.

Extended experiments have been made to test the capability of the adaptive algorithms to solve convection-dominated problems. We refer to [Tro96] and [Cic] for details. We only give here the conclusions we draw from the test cases we computed. We were particularly interested in the dependence of the algorithms on the diffusion parameter ε, on the mesh size, on the position of the interfaces, on the number of subdomains and on the presence of crosspoints in the decomposition.

We notice that the convergence of the ADN and ARN algorithms does not depend on ε, when sufficiently small (which are the cases we are interested in). The convergence is also not sensitive to mesh size. On the contrary, the convergence depends on the position of the interface and on the number of subdomains. Such is the general behaviour for all the algorithms.

The choice of the relaxation parameter is very important. In fact a good choice of θ can accelerate the convergence rate. An analysis of the best value of θ is made in [GGQ]. We observe that, for ADN the number of iterations needed to reach convergence is always lower than for ARN, but for ADN θ changes a lot with different problems, ε, and number of subdomains. On the contrary, for ARN, in the advection dominated cases, the best convergence is always achieved for $\theta = 1$, making this method easier to use.

Figure 1 Numerical solution of the thermal boundary layer problem



For the damped algorithms (d-ADN and d-ARN) the convergence depends on ε, even when small. But the number of iterations to reach convergence is quite small, sometimes twenty times lower than this needed with the ADN or ARN algorithms. The d-ARN algorithm has a bit faster convergence than the d-ADN.

On the two following paragraphs we present two test cases to illustrate the behaviour of the algorithms. The first computation uses a finite element discretization, the second a finite volume/finite element discretization.

Thermal Boundary Layer Problem

This classical benchmark problem, undertakes to find the solution of

$$-\varepsilon\Delta u + 2yu_x = 0 \quad \text{on} \quad \Omega = (0,1) \times (0,0.5)$$

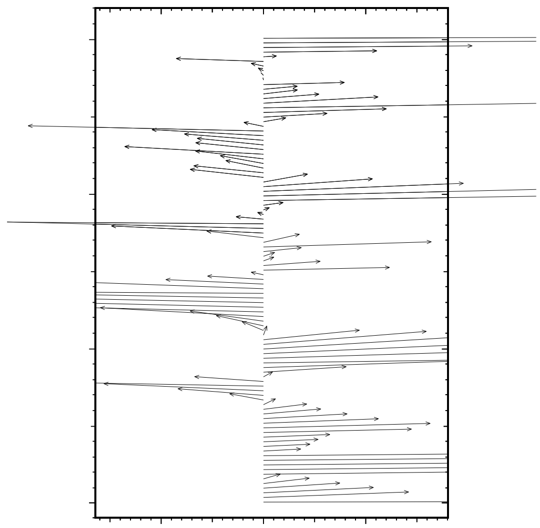
with boundary conditions $u = 1$ for boundary sides $x = 0$ and $y = 0.5$, $u = 0$ for boundary side $y = 0$ and $u = 2y$ for $x = 1$. The problem has two zones of large gradient.

We divide the domain in two subdomains:

$\Omega_1 = (0, 0.7) \times (0, 0.5)$ with 21×41 uniformly spaced grid points

$\Omega_2 = (0.7, 1) \times (0, 0.5)$ with 41×41 uniformly spaced grid points.

Figure 2 Velocity field at an interface



We apply ADN, ARN and d-ARN to problem with $\varepsilon = 10^{-4}$, and we obtain the numerical solution plotted in Fig. 1. The number of iteration needed is 14 for ADN ($\theta = 0.83$), 19 for ARN ($\theta = 1$) and 2 for d-ARN ($\theta = 1$).

Unsteady Calculation — Reservoir Problem

In this section, we show how to extend the algorithms to unsteady calculations. We solve the unsteady advection-diffusion equation:

$$\frac{\partial u}{\partial t} - \varepsilon \Delta u + \operatorname{div}(bu) = 0$$

The first-order Euler implicit scheme applied to the previous problem is:

$$\frac{u^{n+1} - u^n}{\Delta t} - \varepsilon \Delta u^{n+1} + \operatorname{div}(bu^{n+1}) = 0$$

or:

$$\frac{u^{n+1}}{\Delta t} - \varepsilon \Delta u^{n+1} + \operatorname{div}(bu^{n+1}) = \frac{u^n}{\Delta t}$$

We identify the usual formulation by denoting: $a = \frac{1}{\Delta t}$, $\quad f = \frac{u^n}{\Delta t}$. We apply the DD algorithms at each time iteration.

The test case we present now models the saturation of oil in a porous media. A Dirichlet condition ($u = 1$) is imposed on the boundary $y = 0$. On the three other sides, homogeneous Neumann conditions are imposed. The initial conditions are:

$$\begin{cases} u = 1 & \text{on } y = 0 \\ u = 0 & \text{elsewhere.} \end{cases}$$

For this calculation, the viscosity ε is taken equal to 10^{-4}. The velocity field at one of the interfaces can be seen on Fig. 2. We observe that the direction of the velocity changes very often at the interfaces. This is a rather difficult test for the Adaptive DD algorithms.

The entire mesh has 101×101 points. We use a partition in five subdomains. The interfaces are situated at $x = 0.2$, $x = 0.4$, $x = 0.6$, $x = 0.8$, and are parallel to the flow direction. Seven hundred and ninety two points are situated on the interfaces.

At each time step, the d-ADN algorithm requires seven or eight iterations to reach the 10^{-4} convergence stopping criterion. We refer to [Cic] to have more details on the test case solution. By making this calculation, we wanted to know the behaviour of the Adaptive DD algorithms in a real test case in which the interface treatment is particularly complex. And we observed that the d-ADN algorithm is quite efficient and needs a low and constant number of iterations along the time steps.

5 Conclusion

For advection dominated advection-diffusion problems, adaptive methods are certainly superior, because they take into account the direction of the "wind".

By looking at the results obtained, we can conclude that the ADN method is not as robust as the ARN, because of the difficulties arising in the choice of the best parameter θ. But, a theoretical convergence analysis of the ADN algorithm would provide a way to choose the optimal θ and ADN could be more efficient than ARN, in terms of number fo iterations, when ε becomes small. It must be also underlined that ADN performs well even for large ε. The damped versions of the ADN and ARN algorithms accelerate convergence and reduce very significantly the number of iterations needed for a computation. For problems with sufficiently small diffusion, the d-ADN algorithm will provide a faster convergence. And if a discontinuous solution is acceptable, the d-ARN algorithm seems to be the most suited method.

In the first step, we have tested the performances of these adaptive algorithms on an advection-diffusion model problem to analyze their behaviour.

Several developments of our work are possible. One could be the adaptation of the algorithms to the Navier-Stokes equations and to the coupling of the Euler and Navier-Stokes equations. Another could be the use of adaptive methods as preconditioners. And, of course, the development of a mathematical theory would be very helpful.

REFERENCES

[BBF+92] Brezzi F., Bristeau M.-O., Franca L. P., Mallet M., and Rogé G. (1992) A relationship between stabilized finite element methods and the Galerkin method with bubble functions. *Comp. Meths. Appl. Mech. Eng.* 96: 117–129.

[BW86] Bjørstad P. E. and Widlund O. B. (1986) Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.* 23: 1097–1120.

[Cic] Ciccoli M.-C.Adaptive domain decomposition algorithms and finite volume/finite element approximation for advection-diffusion equations. Submitted to Journal of Scientific Computing.

[CQ95] Carlenzoli C. and Quarteroni A. (1995) Adaptive domain decomposition methods for advection - diffusion problems. In Babuska I. e. a. (ed) *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, volume 75 of *IMA Volumes in Mathematics and its Applications*, pages 165–199. Springer Verlag edition.

[GGQ] Gastaldi F., Gastaldi L., and Quarteroni A.Adaptive domain decomposition methods for advection dominated equations. To appear on East-West J. Numer. Math.

[MQ89] Marini L. D. and Quarteroni A. (1989) A relaxation procedure for domain decomposition methods using finite elements. *Numer. Math.* 55: 575–598.

[NR95] Nataf F. and Rogier F. (1995) Outflow boundary conditions and domain decomposition method. In Keyes D. E. and Xu J. (eds) *Proc. Seventh Int. Conf. on Domain Decomposition Meths.*, volume 180 of *Contemporary Mathematics*, pages 289–293. American Mathematical Society edition.

[Qua90] Quarteroni A. (1990) Domain decomposition methods for system of conservation laws: spectral collocation. *SIAM J. Sci. Stat. Comput.* 11: 1029–1052.

[TB95] Tan K. H. and Borsboom M. J. (1995) On generalized Schwarz coupling applied to advection-diffusion problems. In Keyes D. E. and Xu J. (eds) *Proc. Seventh Int. Conf. on Domain Decomposition Meths.*, volume 180 of *Contemporary Mathematics*, pages 289–293. American Mathematical Society edition.

[Tro96] Trotta R. L. (1996) Multidomain finite elements for advection-diffusion equations. *Appl. Numer. Math.* 21: 91–118.