39

ADN and ARN Domain Decomposition Methods for Advection-Diffusion Equations

F. Gastaldi, L. Gastaldi, and A. Quarteroni

1 Introduction

We consider adaptive domain decomposition methods for the solution of advection-diffusion boundary value problems.

The computational domain is partitioned into disjoint subdomains that do not overlap. The original boundary value problem is reformulated in a split form on the subdomains, and the subdomain solutions satisfy suitable matching conditions at subdomain interfaces. These transmission conditions are then used to set up iterative procedures among subdomains. In this work we review a family of methods, known as ADN (Adaptive Dirichlet Neumann) and ARN (Adaptive Robin Neumann), which were previously introduced in [CQ95], [Cic], and [Tro96].

The idea behind these methods is to choose interface conditions which are compatible with those of the hyperbolic problem obtained letting the diffusion coefficient go to zero. Then iterative methods are introduced splitting the above interface conditions in a way which is adapted to the local flow direction. This prevents the rise of artificial layers at subdomain interfaces as the advection becomes dominant. An extensive analysis of the properties enjoyed by these methods is carried out in [GGQ96]. In this work we report briefly the main results of this analysis.

2 Advection-diffusion Boundary Value Problem

Let Ω be a bounded, connected, open subset of $\mathbb{R}^2$ with a Lipschitz continuous boundary $\partial\Omega$. Let $\epsilon > 0$ be a constant diffusion coefficient, $\mathbf{b} = \mathbf{b}(x)$ denote the given flow velocity and $b_0 = b_0(x)$ be an absorption coefficient. The boundary value

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

problem we are considering reads: Find u such that

$$\begin{cases} L_\epsilon u := -\epsilon\Delta u + \operatorname{div}(\mathbf{b}u) + b_0 u & = & f & \text{in } \Omega, \\ & u = & 0 & \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

where $f = f(x)$ is a given body force.

The characteristic quantity $\omega = (2\epsilon)^{-1}\|\mathbf{b}\|_\infty$ (essentially the analog of the Reynolds number for Navier–Stokes equations) will be used; in particular, we will be primarily concerned with the case $\omega \gg 1$ (advection-dominated).

The method can be applied in the case of several subdomains (see Remark 1 below); for simplicity, here we restrict the discussion to the model case where the domain is divided into two non–overlapping subdomains. Let Ω_1 and Ω_2 be these subdomains, whose boundaries $\partial\Omega_1$ and $\partial\Omega_2$ are supposed to be Lipschitz continuous. The common interface $\partial\Omega_1 \cap \partial\Omega_2$ is denoted by Γ; the normal unit vector on Γ pointing into Ω_2 is denoted by $\mathbf{n}$. We assume that Γ is piecewise $\mathbf{C}^1$ and we distinguish three subsets of the regular part of Γ (namely, where $\mathbf{n}$ exists):

$$\begin{cases} \Gamma^0 & := & \{x \in \Gamma : \mathbf{b}(x) \cdot \mathbf{n}(x) = 0\}, \\ \Gamma^{in} & := & \{x \in \Gamma : \mathbf{b}(x) \cdot \mathbf{n}(x) < 0\}, \\ \Gamma^{out} & := & \{x \in \Gamma : \mathbf{b}(x) \cdot \mathbf{n}(x) > 0\}, \end{cases} \tag{2.2}$$

which are identified through the local direction of the flow field $\mathbf{b}(x)$ at the subdomain interface.

The original boundary value problem (2.1) can be reformulated as follows. Denoting by u_1 and u_2 the restriction of the solution u to the subdomains Ω_1 and Ω_2, respectively, it can be shown that u_1 and u_2 satisfy the split problem:

$$L_\epsilon u_1 = f \qquad \text{in} \quad \Omega_1, \tag{2.3}$$

$$L_\epsilon u_2 = f \qquad \text{in} \quad \Omega_2, \tag{2.4}$$

$$u_i = 0 \qquad \text{on} \quad \partial\Omega_i \backslash \Gamma \quad i = 1, 2. \tag{2.5}$$

$$u_1 = u_2 \qquad \text{on} \quad \Gamma, \qquad (D) \tag{2.6}$$

$$\epsilon\frac{\partial u_1}{\partial \mathbf{n}} = \epsilon\frac{\partial u_2}{\partial \mathbf{n}} \qquad \text{on} \quad \Gamma. \qquad (N) \tag{2.7}$$

The interface matching conditions (2.6) and (2.7) enforce the simultaneous continuity of the subdomain solutions (Dirichlet condition, say D) and of their normal derivatives (Neumann condition, say N). Besides the *Dirichlet–Neumann* formulation of the transmission conditions one could use as well the continuity of the fluxes (Robin condition, say R) and combine it either with D or with N. For example (2.6) and (2.7) can be replaced equivalently by the *Robin–Neumann* matching conditions:

$$\epsilon\frac{\partial u_1}{\partial \mathbf{n}} - \mathbf{b}\cdot\mathbf{n} u_1 = \epsilon\frac{\partial u_2}{\partial \mathbf{n}} - \mathbf{b}\cdot\mathbf{n} u_2 \quad \text{on} \quad \Gamma, \qquad (R) \tag{2.8}$$

$$\epsilon\frac{\partial u_1}{\partial \mathbf{n}} = \epsilon\frac{\partial u_2}{\partial \mathbf{n}} \quad \text{on} \quad \Gamma, \qquad (N) \tag{2.9}$$

provided $meas\{x \in \Gamma : \mathbf{b}(x) \cdot \mathbf{n}(x) = 0\} = 0$.

For brevity, we restrict our discussion to these two types of transmission conditions. For the analysis of other conditions we refer to [GGQ96].

3 Iterative Algorithms for Domain Decomposition

The next step is to set up iterative procedures between the subdomains, based on the problem splitting (2.3)–(2.5) with D-N ((2.6)–(2.7)) or R-N ((2.8)–(2.9)) interface conditions. We define a sequence $\{u_1^n, u_2^n\}$, where u_1^n satisfies (2.3) and (2.5), u_2^n satisfies (2.4) and (2.5) along with either type of boundary conditions at the subdomain interface Γ. A first option is given by the standard Dirichlet–Neumann iterative algorithm: Given u_i^0 in Ω, $(i = 1, 2)$ solve for each $n \geq 1$

$$(D)\left\{\begin{array}{rcll} L_\epsilon u_1^n & = & f & \text{in } \Omega_1 \\ u_1^n & = & 0 & \text{on } \partial\Omega_1\backslash\Gamma \\ u_1^n & = & \lambda^{n-1} & \text{on } \Gamma \end{array}\right. \qquad (N)\left\{\begin{array}{rcll} L_\epsilon u_2^n & = & f & \text{in } \Omega_2 \\ u_2^n & = & 0 & \text{on } \partial\Omega_2\backslash\Gamma \\ \epsilon\frac{\partial u_2^n}{\partial \mathbf{n}} & = & \epsilon\frac{\partial u_1^n}{\partial \mathbf{n}} & \text{on } \Gamma \end{array}\right. ,$$

where $\lambda^{n-1} = \theta u_2^{n-1} + (1-\theta)u_1^{n-1}$ and θ is a relaxation parameter.

Of course, the conditions at the interface can be interchanged; hence one can also solve first the problem in Ω_1 with Neumann condition at the interface and then the problem in Ω_2 with Dirichlet conditions at the interface. The two choices are suitable for viscous-dominated flows where the skew symmetric part of the operator is not too big. Well known results on such schemes can be found in the papers by Bjørstad and Widlund [BW86] and by Marini and Quarteroni [MQ89], where relaxation is proven to be essential for convergence. But for advection-dominated problems the conditions at the interface have to be set up carefully.

Indeed, the interface conditions have to be adapted to the orientation of the transport field across the interface. The reason why Neumann conditions are used on outflow boundaries is that a Dirichlet condition prescribing specific values for the solution could generate artificial internal layers whose steepness is proportional to ω. Similar comments hold also for Robin–Neumann matching conditions with Robin playing the same role as Dirichlet. Therefore we consider the following algorithm: given u_i^0 in Ω_i, $(i = 1, 2)$, solve for each $n \geq 1$

$$\left\{\begin{array}{rcll} L_\epsilon u_1^n & = & f & \text{in } \Omega_1 \\ u_1^n & = & 0 & \text{on } \partial\Omega_1\backslash\Gamma \\ \psi(u_1^n) & = & \lambda^{n-1} & \text{on } \Gamma^{in} \\ \epsilon\frac{\partial u_1^n}{\partial \mathbf{n}} & = & \epsilon\frac{\partial u_2^{n-1}}{\partial \mathbf{n}} & \text{on } \Gamma^{out} \end{array}\right. \quad \left\{\begin{array}{rcll} L_\epsilon u_2^n & = & f & \text{in } \Omega_2 \\ u_2^n & = & 0 & \text{on } \partial\Omega_2\backslash\Gamma \\ \psi(u_2^n) & = & \mu^n & \text{on } \Gamma^{out} \\ \epsilon\frac{\partial u_2^n}{\partial \mathbf{n}} & = & \epsilon\frac{\partial u_1^n}{\partial \mathbf{n}} & \text{on } \Gamma^{in} \end{array}\right. , \tag{3.10}$$

with

$$\psi(v) := \left\{\begin{array}{ll} v & \text{for ADN method} \\ \epsilon\frac{\partial v}{\partial \mathbf{n}} - \mathbf{b}\cdot\mathbf{n}v & \text{for ARN method} \end{array}\right. , \tag{3.11}$$

$$\lambda^{n-1} := \theta'\psi(u_2^{n-1}) + (1-\theta')\psi(u_1^{n-1}), \tag{3.12}$$

$$\mu^n := \theta''\psi(u_1^n) + (1-\theta'')\psi(u_2^{n-1}), \tag{3.13}$$

θ' and θ'' being two real parameters that are used to accelerate the convergence of the iterative procedure. Notice that the condition on the interface is now split into two parts: Along the outflow part of the interface we enforce the continuity of the normal derivative, while along the inflow part we enforce the continuity of the trace for Adaptive Dirichlet–Neumann algorithm (ADN) or the continuity of the flux for Adaptive Robin–Neumann algorithm (ARN).

The parameters θ' and θ'' in (3.12) and (3.13) are used to guarantee possible *under-relaxation* when needed to ensure convergence. Typically, a single parameter θ $(= \theta' = \theta'')$ suffices (two parameters allow better flexibility in pursuing an optimal criterion), and often $\theta = 1$ (no relaxation) is a very good choice.

Finally, let us briefly introduce the damped version of the Adaptive Robin–Neumann method (denoted by d–ARN). It consists in substituting the condition that enforces the continuity of the normal derivative on Γ^{out} with a homogeneous Neumann condition. Hence the d–ARN algorithm consists in solving the sub-problem in Ω_1 provided the condition along the outflow part of Γ is replaced by the following equation: $\epsilon \partial u_1^n / \partial \mathbf{n} = 0$ on Γ^{out}. Similarly, one solves the sub– problem in Ω_2 with the damped condition: $\epsilon \partial u_2^n / \partial \mathbf{n} = 0$ on Γ^{in}. The main reason for introducing the damped form of our algorithms is that the damped forms weaken the coupling between u_1^n and u_2^n at the interface Γ, so that the convergence of the corresponding algorithm is faster in general. In particular, when the flow field has a constant direction at the subdomain interface, i.e., $\mathbf{b}(x) \cdot \mathbf{n}(x)$ is either always positive or always negative on Γ, then a single iteration is enough to solve the given problem. This introduces an error, which can be proved not to grow at each iteration, so that the method produces a sequence which is weakly convergent. Moreover, the solution we get for n going to infinity is not too far from the exact one and the error can be measured in terms of a suitable norm of the normal derivative of the exact solution along the interface multiplied by $\sqrt{\epsilon}$. Therefore if the interface is far from any layer (hence the normal derivative is bounded independently of ϵ), then we obtain an error bound of order $\sqrt{\epsilon}$.

The convergence of these methods can be proven working out the error equations, for the complete analysis see [GGQ96]. By subtracting the iterative solution at step n from the exact solution we obtain the errors $e_1^n = u_1 - u_1^n$ and $e_2^n = u_2 - u_2^n$, which solve the same problems as before with homogeneous data inside Ω_1 and Ω_2, respectively.

For the ARN method without relaxation we have obtained the following estimate

$$\|e_2^n\|_\Gamma \leq \|e_2^{n-1}\|_\Gamma \quad \forall n, \tag{3.14}$$

where

$$\|e_2^n\|_\Gamma^2 = \int_{\Gamma^{in}} \frac{1}{|\mathbf{b}\cdot\mathbf{n}|} \left(\epsilon \frac{\partial e_2^n}{\partial \mathbf{n}} - \mathbf{b}\cdot\mathbf{n} e_2^n \right)^2 ds \quad + \int_{\Gamma^{out}} \frac{1}{\mathbf{b}\cdot\mathbf{n}} \left(\epsilon \frac{\partial e_2^n}{\partial \mathbf{n}} \right)^2 ds. \tag{3.15}$$

The formula (3.14) expresses that the error at the interface, measured in the norm (3.15), does not grow at each iteration. Moreover, the estimate (3.14) implies a weak convergence of the sequence $\{u_1^n, u_2^n\}$ as it is stated in the following lemma:

Lemma 1 *The sequence* $\{u_1^n, u_2^n\}$ *converges weakly in* $\mathbf{H}^1(\Omega_1) \times \mathbf{H}^1(\Omega_2)$ *to* $\{u_1, u_2\}$ *solution of (2.3)–(2.5), (2.8)–(2.9).*

However, this result cannot provide any useful information for example on the speed of the convergence. Therefore we will detail our analysis in a sample problem (see next section).

Remark 1 *An extensive experimental analysis of these methods in the framework of several kinds of numerical realizations (finite elements, finite volumes, spectral methods) and of decompositions using more than two subdomains, including crosspoints, is carried out in [CQ95], [Cic], and [Tro96]. Obviously, when dealing with more than two subdomains, the Robin or Neumann conditions have to be imposed along each interface according to the local direction of the transport field. The analog of the estimate (3.14) can be still obtained, no matter whether there are crosspoints or not. The iterative algorithm can be performed similarly in the ADN framework.*

4 Analysis of a Two-dimensional Case with Constant Transport

Let Ω be the unit square $]0,1[\times]0,1[$, divided into $\Omega_1 =]0,\gamma[\times]0,1[$, $\Omega_2 =]\gamma,1[\times]0,1[$, where $\gamma \in]0,1[$ is given. The interface is the set $\Gamma = \{\gamma\}\times]0,1[$. The transport field is $\mathbf{b} = (b,0)$, with b a positive constant and we take b_0 a non-negative constant. Hence the problem reads:

$$\left\{ \begin{array}{rcll} L_\epsilon u := -\epsilon\Delta u + bu_x + b_0 u & = & f & \text{in } \Omega \\ u & = & 0 & \text{on } \partial\Omega. \end{array} \right. \tag{4.16}$$

We construct the sequence $\{u_1^n, u_2^n\}$ as in (3.10)–(3.13) keeping in mind that Γ^{out} coincides with Γ. Then introducing as before the errors $e_i^n = u_i - u_i^n$, $i = 1,2$, we construct a sequence in the following way: Given a function g_1^{n-1}, solve

$$(N) \left\{ \begin{array}{rcll} L_\epsilon e_1^n & = & 0 & \text{in } \Omega_1 \\ e_1^n & = & 0 & \text{on } \partial\Omega_1 \backslash \Gamma \\ \epsilon e_{1x}^n & = & \epsilon g_1^{n-1} & \text{on } \Gamma, \end{array} \right.$$

then set $g_2^n = \theta\psi(e_1^n) + (1-\theta)\psi(e_2^{n-1})$ on Γ and solve

$$(R) \text{ or } (D) \left\{ \begin{array}{rcll} L_\epsilon e_2^n & = & 0 & \text{in } \Omega_2 \\ e_2^n & = & 0 & \text{on } \partial\Omega_2 \backslash \Gamma \\ \psi(e_{2x}^n) & = & g_2^n & \text{on } \Gamma \end{array} \right. ,$$

and finally, set $g_1^n = e_{2x}^n$ on Γ.

When $\theta = 1$ (no relaxation), then by separation of variables we obtain

$$g_1^n(y) = \sum_{k=1}^{\infty} \eta_k^n Y_k(y) \quad \forall n \geq 0, \tag{4.17}$$

where Y_k are the eigenfunctions of the induced spectral problem with respect to y: $-\epsilon Y_k'' = \lambda_k Y_k$ in $(0,1)$, $Y_k(0) = Y_k(1) = 0$.

The coefficients of the expansion (4.17) satisfy a recursive formula

$$\eta_k^n = \rho_k \eta_k^{n-1}, \tag{4.18}$$

with ρ_k given by

$$\rho_k^{ARN} = \frac{\tau_k \coth(\tau_k\gamma) - \omega}{\tau_k \coth(\tau_k\gamma) + \omega} \, \frac{\tau_k \coth(\tau_k(1-\gamma)) - \omega}{\tau_k \coth(\tau_k(1-\gamma)) + \omega},$$

$$\rho_k^{ADN} = -\frac{\tau_k \coth(\tau_k(1-\gamma)) - \omega}{\tau_k \coth(\tau_k \gamma) + \omega},$$

and $\omega = b/(2\epsilon)$, $\tau_k = \sqrt{b^2 + 4\epsilon(b_0 + \lambda_k)}/(2\epsilon)$.

If $\theta \neq 1$ then the recursive formula (4.18) holds with ρ_k replaced by $\rho_k^\theta = 1-\theta(1-\rho_k)$. To have the convergence of the method one should show that the sequence g_1^n converges to 0, but the convergence of g_1^n corresponds to the convergence of the sequences $\eta^n := \{\eta_k^n\}_{k\geq 1}$ in the Hilbert space ℓ^2. Owing to (4.18) it turns out that a sufficient condition is that

$$\sup_k |\rho_k| < 1 \tag{4.19}$$

It is not difficult to prove that $0 < \rho_k^{ARN} < 1$ for all k. Moreover, ρ_k^{ARN} is an increasing function of k and converges to 1 as $k \to +\infty$. On the other hand ρ_k^{ADN} is negative. Moreover for γ not too close to 1 we obtain that $-1 < \rho_k^{ADN} < 0$ for all k and the function ρ_k^{ADN} is decreasing and converges to -1 as $k \to +\infty$.

In Fig. 1 the graphs of ρ_k^{ARN} as a function of k for different values of ϵ are drawn (the solid line corresponds to $\epsilon = 10^{-2}$, the dotted line to $\epsilon = 10^{-3}$, the dashed-dotted line to $\epsilon = 10^{-4}$, the dashed line to $\epsilon = 10^{-5}$). Fig. 2 shows the graphs of ρ_k^{ADN} when γ is not too big, for the same values of ϵ.

Fig.1 ρ_k^{ARN} for several values of ϵ

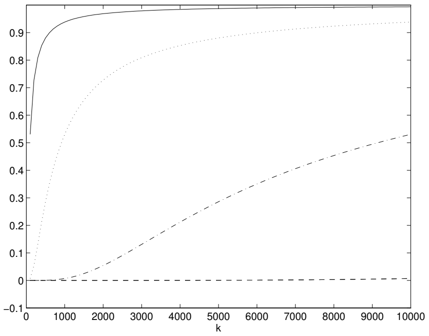
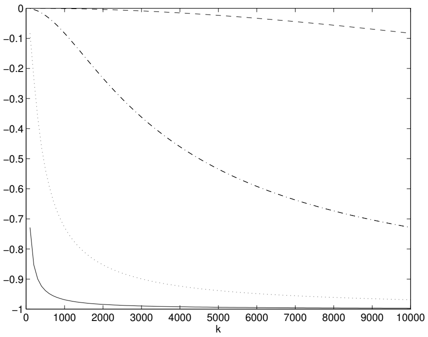


Fig.2 ρ_k^{ADN} for several values of ϵ



Notice that both methods ARN and ADN do not fulfill the sufficient condition (4.19) for convergence. A cure could be the use of a relaxation strategy. It is easy to see that for ARN method ρ_k^θ converge to 1 as $k \to +\infty$, while for ADN it is possible to choose the parameter θ in order to make the sequence converge. However, the formulas and the pictures show that if the high frequency modes are neglected then the two methods provide good convergence for $\theta = 1$. This is in agreement with the numerical results presented in [Tro97]. As a matter of fact a finite dimensional approximation uses only a finite number of modes, as will be shown in the next section.

5 Approximation by Finite Elements

The methods discussed in Sections 2 and 3 are naturally rephrased in a weak form (see [GGQ96]), which is most convenient for finite element approximations. When there are several subdomains, crosspoints may show up. In this case, the matching conditions involving derivatives are naturally enforced through the test functions associated with the crosspoints (see [CQ95, Tro96], where extensive numerical validation is carried out).

The convergence analysis developed in Section 4 can be adapted to the finite element approximation using bilinear finite elements. Let us define the discrete iteration errors e^n_{1h} and e^n_{2h} as the difference between the discrete one–domain solution and the n-th discrete iterate. These errors can be written, by separation of variables, as sums of tensor products

$$e^n_{1h} = \sum_{k=1}^{N_y} \mu^{(n)}_{kh} X_{kh}(x) Y_{kh}(y), \quad e^n_{2h} = \sum_{k=1}^{N_y} \eta^{(n)}_{kh} Z_{kh}(x) Y_{kh}(y),$$

where Y_{kh} are the piecewise linear eigensolutions in the y variable, X_{kh} and Z_{kh} are the piecewise linear solutions of the associated problems in the x variable on $(0,\gamma)$ and $(\gamma,1)$, respectively. Then the coefficients of these linear combinations satisfy a recursive relation $\eta^{(n)}_{kh} = \rho_{kh}\eta^{(n-1)}_{kh}$, with the reduction factor satisfying $0 < \rho_{kh} < 1$. Hence the sufficient condition (4.19) is satisfied and the discrete iterative procedure converges. In this case the introduction of a relaxation strategy can improve the speed of convergence with a suitable choice of the relaxation parameter.

6 Conclusions

For advection-dominated problems, the adapted iterative algorithms presented have good convergence properties, when a finite number of modes are taken into account along the interface. We observe that this happens when finite dimensional discretizations of the problem are considered. The damped version of the methods is very efficient, with a reasonable choice for the location of the interface.

Due to space limitations, we do not address the issue of efficient implementation (the interested reader can refer to [GGQ96], sect. 1.4): in particular, when using a very large number of subdomains, a coarse grid solver (based on the same adaptive principle) is required in order that the algorithm is scalable.

Acknowledgement

The third author acknowledges the financial support of the Sardinian Regional Authorities.

REFERENCES

[BW86] Bjørstad P. 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.

[GGQ96] Gastaldi F., Gastaldi L., and Quarteroni A. (1996) Adaptive domain decomposition methods for advection dominated equations. *East-West J. Numer. Math.* 4: 165–206.

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

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

[Tro97] Trotta L. (1997) Multidomain finite volumes and finite elements for advection–diffusion equations. In Bjørstad P., Espedal M., and Keyes D. (eds) *DD9 Proceedings*.