Non Overlapping Domain Decomposition for Evolution Operators

Jacques-Louis Lions *and Olivier Pironneau†

April 10, 2000

Abstract

In this note we study the domain decomposition of evolution problems into subproblems on non overlapping subdomains. The method is similar to the one used in a previous note [4] for overlapping subdomains. It used Lagrange multipliers without referring to an optimization problem so as to avoid two point boundary value problems.

Résumé

Nous étudions dans cette note, qui poursuit la note [4], la décomposition de domaine pour des opérateurs d'évolution, la décomposition est sans recouvrement. La méthode présentée ici est analogue à [4], mais avec des détails techniques différents. On utilise des multiplicateurs de Lagrange qui peuvent être considérés comme correspondants à des "feedback" a priori pour des contrôles virtuels.

1 Introduction

After considering in [4],the domain decomposition methods for evolution operators in the overlapping case, we consider in this note the *non overlapping situations*, always with the idea of "*Virtual Control with a priori feedback*".

2 The Continuous Problem and the Domain Decomposition

Let Ω be an open set in R^d. with smooth boundary Γ. We use the following notations:

$$V = H_0^1(\Omega), \quad H = L^2(\Omega), \quad (f,g) = \int_\Omega fg dx$$

*Académie des Sciences

†Université Pierre et Marie Curie et IUF

$$c(f,g) = \int_\Omega \gamma(x) f g dx, \quad 0 < \gamma_0 \leq \gamma(x), \quad \gamma \in L^\infty(\Omega),$$
$$a(u,\hat{u}) = \sum \int_\Omega a_{kl} \frac{\partial u}{\partial x_l} \frac{\partial \hat{u}}{\partial k} dx \quad \forall u, \hat{u} \in V$$

where $a_{kl} \in L^\infty(\Omega)$,

$$a_{kl}\xi_k\xi_l \geq \alpha \sum_k \xi_k^2, \quad \alpha > 0, \quad \text{a.e. in } \Omega.$$

The continuous problem in variational form (up to now it is the same situation than in the previous note) is to find u such that

$$u \in L^2(V) \cap L^\infty(H) \quad (\text{ where } L^p(E) = L^p(0,T;E),\ T, \text{ fixed}) \tag{1}$$
$$c(\frac{\partial u}{\partial t}, \hat{u}) + a(u,\hat{u}) = (f,\hat{u}) \quad \forall \hat{u} \in V \quad \text{ with } f \text{ given in } L^2(V') \tag{2}$$
$$u(0) = 0 \tag{3}$$

(Weak solutions are considered as in Remark 1 of [4]).

Problem (1)(2)(3) has a unique solution, that we want to approximate.

We assume that the domain Ω is made of two parts Ω_1,Ω_2 and denote by S their common boundary and by Γ_i the intersection of $\partial\Omega$ with $\partial\Omega_i$,

$$\begin{array}{l} \overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2 \quad \Omega_1 \cap \Omega_2 = \emptyset \\ S = \partial\Omega_1 \cap \partial\Omega_2 \quad \Gamma_i = \partial\Omega \cap \partial\Omega_i \end{array} \tag{4}$$

i.e. a non overlapping decomposition of Ω.
We are going to use this domain decomposition with the idea that the coefficients of $c(u,\hat{u})$ and of $a(u,\hat{u})$ may be of different orders of magnitude in Ω_1 and in Ω_2. Let us introduce the following notations:

$$V_i = \{v_i \mid v_i \in H^1(\Omega_i), \quad v_i = 0 \ \text{ on } \ \Gamma_i, \quad v_i|_S \text{ arbitrary}\} \quad H_i = L^2(\Omega_i),$$
$$c_i(u_i,\hat{u}_i) = \int_{\Omega_i} \gamma(x) u_i \hat{u}_i dx,$$
$$a(u_i,\hat{u}_i) = \sum \int_{\Omega_i} a_{kl} \frac{\partial u_i}{\partial x_l} \frac{\partial \hat{u}_i}{\partial x_k} dx \quad \forall u, \hat{u} \in V.$$

There exist γ_i and α_i such that

$$\left| \begin{array}{l} c_i(u_i,u_i) \geq \int_{\Omega_i} \gamma_i u_i^2 dx = \gamma_i \|u_i\|^2_{H_i} \\ a_i(u_i,u_i) \geq \alpha_i \|u_i\|^2_{V_i} \end{array} \right. \quad \forall u_i \in H_i \tag{5}$$

We assume that the γ_i's and the α_i's are the best possible constants (or close to the best one) *so that these values reflect the possibly different order of magnitude of the coefficients in* Ω_1 *and* Ω_2.
We introduce now (in section 2) the continuous decomposition method (in infinite dimension) and we proceed (in section 3) with the numerical algorithms.

3 The infinite dimension decomposition approximation

We shall set

$$(f,g)_S = \int_S f g dS,$$

the scalar product of f, g in $L^2(S)$. We also introduce $H_0^1(S)$, i.e. the space of elements $\phi \in L^2(S)$ such that all first order tangential derivatives of ϕ are in $L^2(S)$, i.e. if $d=2$, $\frac{\partial \phi}{\partial \sigma} \in L^2(S)$ and if $d=3$, $\frac{\partial \phi}{\partial \sigma_1}, \frac{\partial \phi}{\partial \sigma_2} \in L^2(S)$ (where σ_1, σ_2 are the tangent vector fields on S) and which are such that $\phi = 0$ on ∂S. We introduce

$$\left| \begin{array}{c} b(\phi, \hat{\phi}) = \text{continuous bilinear form on } H_0^1(S), \\ \textit{coercive, symmetric, chosen as simple as possible or even more simply} \\ b(\phi,\hat{\phi}) \textit{symmetric, coercive on } L^2(S), \textit{ a choice we make for the following} \end{array} \right. \tag{6}$$

Remark 1 *In a sense, since the traces of V_i on S are in $H^{1/2}(S)$ (in fact in $H_{00}^{1/2}(S)$ – cf LIONS and MAGENES[7] for these function spaces), it would be more "natural" but much more complicated to introduce $b(\phi,\hat{\phi})$ with the same properties as (6) but on $H_{00}^{1/2}(S)$ rather than $H_0^1(S)$. Another choice (of course a simpler one) would be to consider b on $L^2(S)$ but this has a drawback from the numerical performance point of view.*

We introduce now the following decomposed system

$$\left| \begin{array}{l} c_1(\frac{\partial u_1}{\partial t}, \hat{u}_1) + a_1(u_1, \hat{u}_1) + (\hat{u}_1, \lambda)_S = \int_{\Omega_1} f \hat{u}_1 dx \quad \forall \hat{u}_1 \in V_1, \\ c_2(\frac{\partial u_2}{\partial t}, \hat{u}_2) + a_2(u_2, \hat{u}_2) - (\hat{u}_2, \lambda)_S = \int_{\Omega_2} f \hat{u}_2 dx \quad \forall \hat{u}_2 \in V_2, \\ \epsilon(\frac{\partial \lambda}{\partial t}, \hat{\lambda})_S + \eta b(\lambda, \hat{\lambda}) - (u_1 - u_2, \hat{\lambda})_S = 0 \quad \forall \hat{\lambda} \in H_0^1(S), \end{array} \right. \tag{7}$$

where

$$\left| \begin{array}{c} u_i \in L^2(V_i) \cap L^\infty(H_i), \\ \lambda \in L^2(H_0^1(S)) \cap L^\infty(L^2(S)), \end{array} \right. \tag{8}$$

$$u_i(0) = 0, \quad \lambda(0) = 0 \tag{9}$$

and where in $(7)_3$, ϵ and η are given > 0 and "small".

Problem (7)(8)(9) admits a unique solution: $u_i = u_{i\epsilon\eta}$, $\lambda = \lambda_{\epsilon\eta}$.

One can prove that

$$\epsilon, \eta \to 0 \quad \Rightarrow \quad u_{i\epsilon\eta} \to u_i \text{ in } L^2(V_i) \text{ weakly and in } L^\infty(H_i) \text{ weak star} \tag{10}$$

$$\text{where} \quad u_i = \text{ restriction of the solution of (1)(2)(3)to } \Omega_i. \tag{11}$$

We introduce now algorithms for the numerical approximations of (7)(8)(9).

4 The numerical approximation algorithms

We introduce the time step δt and we denote by u_i^n, λ^n the approximations of u_i, λ at time $n\delta t$.
Exactly as in [4] we have 3 possibilities, depending if we compute λ^{n+1} first and u_i^{n+1} next (cf algorithm (I)) or the other way around (algorithm (II)) or if we compute λ^{n+1} and u_i^{n+1} in parallel (Algorithm (III)).
Denote $H = H_0^1(S)$.

Assuming that λ^n and u_i^n are known, we define $\lambda^{n+1},\ u^{n+1}$ by

Algorithm (I)

$$\left|\begin{array}{l} \epsilon(\frac{\lambda^{n+1}-\lambda^n}{\delta t}, \hat{\lambda})_S + \eta b(\lambda^{n+1}, \hat{\lambda}) - (u_1^n - u_2^n, \hat{\lambda})_S = 0 \qquad \forall \hat{\lambda} \in H, \\ c_1(\frac{u_1^{n+1}-u_1^n}{\delta t}, \hat{u}_1) + a_1(u_1^{n+1}, \hat{u}_1) + (\hat{u}_1, \lambda^{n+1})_S = \int_{\Omega_1} f^n \hat{u}_1 dx \quad \forall \hat{u}_1 \in V_1, \\ c_2(\frac{u_2^{n+1}-u_2^n}{\delta t}, \hat{u}_2) + a_2(u_2^{n+1}, \hat{u}_2) - (\hat{u}_2, \lambda^{n+1})_S = \int_{\Omega_2} f^n \hat{u}_2 dx \quad \forall \hat{u}_2 \in V_2, \end{array}\right. \tag{12}$$

Algorithm (II)

$$\left|\begin{array}{l} c_1(\frac{u_1^{n+1}-u_1^n}{\delta t}, \hat{u}_1) + a_1(u_1^{n+1}, \hat{u}_1) + (\hat{u}_1, \lambda^n)_S = \int_{\Omega_1} f^n \hat{u}_1 dx \quad \forall \hat{u}_1 \in V_1, \\ c_2(\frac{u_2^{n+1}-u_2^n}{\delta t}, \hat{u}_2) + a_2(u_2^{n+1}, \hat{u}_2) - (\hat{u}_2, \lambda^n)_S = \int_{\Omega_2} f^n \hat{u}_2 dx \quad \forall \hat{u}_2 \in V_2, \\ \epsilon(\frac{\lambda^{n+1}-\lambda^n}{\delta t}, \hat{\lambda})_S + \eta b(\lambda^{n+1}, \hat{\lambda}) - (u_1^{n+1} - u_2^{n+1}, \hat{\lambda})_S = 0 \quad \forall \hat{\lambda} \in H, \end{array}\right. \tag{13}$$

Algorithm (III)

$$\left|\begin{array}{l} \epsilon(\frac{\lambda^{n+1}-\lambda^n}{\delta t}, \hat{\lambda})_S + \eta b(\lambda^{n+1}, \hat{\lambda}) - (u_1^n - u_2^n, \hat{\lambda})_S = 0 \quad \forall \hat{\lambda} \in H, \\ c_1(\frac{u_1^{n+1}-u_1^n}{\delta t}, \hat{u}_1) + a_1(u_1^{n+1}, \hat{u}_1) + (\hat{u}_1, \lambda^n)_S = \int_{\Omega_1} f^n \hat{u}_1 dx \quad \forall \hat{u}_1 \in V_1, \\ c_2(\frac{u_2^{n+1}-u_2^n}{\delta t}, \hat{u}_2) + a_2(u_2^{n+1}, \hat{u}_2) - (\hat{u}_2, \lambda^n)_S = \int_{\Omega_2} f^n \hat{u}_2 dx \quad \forall \hat{u}_2 \in V_2, \end{array}\right. \tag{14}$$

where for instance $f^n = (\int_{n\delta t}^{(n+1)\delta t} f dt)/\delta t$.
To discretize in space it suffices to replace V_i by V_{ih} the finite element space of piecewise linear continuous functions on the triangulation of Ω restricted to Ω_i and to replace H by H_h the space of continuous piecewise linear functions on the segmentation of S which is the trace of the triangulation of Ω.

An analysis of the stability of these algorithms leads to the following results. For algorithm (I) there is a stability condition

$$\delta t \leq c\eta \tag{15}$$

where c is a constant which depends on the forms c_1, c_2 and b.
For algorithm (II) there is stability if

$$\delta t \leq c\epsilon \tag{16}$$

where c is a constant which depends on the forms a_1 and a_2 For algorithm(III) thee is stability if both (15) and (16) hold true.

5 Extensions and variants

1. The previous algorithm can be modified to have different time steps δt_1 and δt_2 in the different regions if desired.

2. The previous methods and results readily extend to other boundary conditions, to systems of equations and to higher order equations.

3. They also extend to the case of an arbitrary non-overlapping decomposition $\overline{\Omega} = \cup \overline{\Omega}_i$.

4. The results also extend to non-linear problems. The non-overlapping decompositions do *not* lead to the difficulties which seem to be present, for non-linear problems, in the overlapping decompositions.

5. One can add to (7) (and therefore to the various algorithms) terms which are aimed at "increasing" the quality of the adjustment at the interface S between u_1 and u_2.
   To this effect let $d(\phi, \hat{\phi})$ be a continuous bilinear form on $H^1(S)$ (resp $L^2(S)$) which is symmetric, coercive on $H^1(S)$ (resp $L^2(S)$) and which is chosen as simple as possible. We consider then the system of equations (compare to (7)).

$$\left| \begin{array}{c} c_1(\frac{\partial u_1}{\partial t}, \hat{u}_1) + a_1(u_1, \hat{u}_1) + d(u_1 - u_2, \hat{u}_1) + (\hat{u}_1, \lambda)_S = \int_{\Omega_1} f \hat{u}_1 dx \\ c_2(\frac{\partial u_2}{\partial t}, \hat{u}_2) + a_2(u_2, \hat{u}_2) - d(u_1 - u_2, \hat{u}_2) - (\hat{u}_2, \lambda)_S = \int_{\Omega_2} f \hat{u}_2 dx \\ \forall \hat{u}_i \in W_i = \{w \in V_i \ : \quad w|_S \in H^1(S)\}, \ i = 1, 2. \\ \epsilon(\frac{\partial \lambda}{\partial t}, \hat{\lambda})_S + \eta b(\lambda, \hat{\lambda}) - (u_1 - u_2, \hat{\lambda})_S = 0 \quad \forall \hat{\lambda} \in H_0^1(S), \end{array} \right. \tag{17}$$

where u_i, λ satisfy (8)(9) and

$$(u_1 - u_2)|_S \in L^2(0, T; H^1(S)) \tag{18}$$

Remark 2 *In fact one can replace everywhere above $H^1(S)$ by $H_0^1(S)$*

One can introduce now numerical algorithms as before. For instance, the analogous of Algorithm(II) is as in (13) with the addition in $(13)_i$ of terms

$$d(u_i^{n+1} - u_j^{n+1}, \hat{u}_i), \quad i, j = 1, 2, \quad j \neq i. \tag{19}$$

Under condition (16) we do have stability and moreover

$$\left| \begin{array}{c} \sum_{n=0}^{N} (d(u_1^{n+1} - u_2^n) + d(u_1^n - u_2^{n+1})), \quad N\delta t = T, \\ \text{remains bounded as } N \to \infty. \end{array} \right. \tag{20}$$

Remark 3 *A more standard algorithm with a penalty tterm is obtained by removing the λ's.*

$$c_i(\frac{\partial u_i}{\partial t}, \hat{u}_i) + a_i(u_i, \hat{u}_i) + (-1)^i \rho d(u_2 - u_1, \hat{u}_i) = \int_{\Omega_i} f \hat{u}_i dx, \forall \hat{u}_i \in W_i \tag{21}$$

where $\rho > 0$. If $\rho \to \infty$ (penalty term) one has an approximation of the solution. One obtains a stable scheme by using (19).

6 Another variant

Let us now think of λ in (7) as the *trace* on S of $\lambda \in H^1(\Omega_1)$ (in fact with $\lambda = 0$ on Γ_1, so that $\lambda|_S \in H^1_{00}(S)$).
Actually in what follows we could as well think of λ as the trace on S of $\lambda \in H^1_0(\Omega_2)$, or also as the trace of $\lambda \in H^1_0(\Omega)$.
Let $b(\lambda, \hat{\lambda})$ be now a continuous bilinear form on $H^1(\Omega_1)$, symmetric and as simple as possible. We then replace $(7)_3$ by

$$\epsilon(\frac{\partial \lambda}{\partial t}, \hat{\lambda})_{\Omega_1} + \eta b(\lambda, \hat{\lambda}) - (u_1 - u_2, \hat{\lambda})_S = 0 \quad \forall \hat{\lambda} \in V_1, \quad \lambda \in L^2(0, T, V_1). \quad (22)$$

We use then the analogous of algorithm(I). We obtain similar stability conditions, the algorithms being simpler to implement numerically.

7 Numerical Example

Convection-diffusion problems in medium with very large variations of diffusion and dissipation constants can be a numerical challenge because the physical time scales are very different from one region to the next. The method described here is well suited and will allow different time steps in the different regions.

We consider the convection-diffusion equation

$$\partial_t u + \mathbf{v} \cdot \nabla u - \nabla \cdot (\nu \nabla u) = 0 \ \text{ in } \ \Omega \times (0, T)$$

with zero initial and boundary conditions, except at the right boundary where an homogeneous Neumann condition is applied.

The problem is in $\Omega = (0, 1) \times (0, 2)$. It is an academic example of the dissipation of a pollutant from an enclosure C into a medium Ω_1 (rock) with low diffusion but cracked (the vertical boundary next to the circle on figure 1). Furthermore below Ω_1 there is another medium Ω_2 (sand) with large diffusion constant ν_2; there the pollutant is also convected (water in sand) at velocity $\mathbf{v}$. The velocity derives from a potential ϕ solution of

$$-\nabla \cdot (\mu \nabla \phi) = 0 \ \text{ in } \ \Omega_2 \quad \phi|_{x=0} = 0 \quad \phi|_{x=1} = 1 \text{ and } \frac{\partial \phi}{\partial n} = 0 \text{ elsewhere}$$

and $\mathbf{v} = -\mu \nabla \phi$.

As described above the equations are discretized in time by an implicit Euler scheme and in space by the finite element method of degree one on triangles. The convection term is treated by the Galerkin-Characteristic method[5] and $X(x)$ denotes an approximation of $x - \mathbf{v}\delta t$.

We have chosen the following Domain Decomposition Method:

$$\frac{1}{\delta t}(u_i^{n+1} - u_i^n \mathrm{o} X, \hat{u}_i) + (\nu_i \nabla u_i^{n+1}, \nabla \hat{u}_i)$$
$$+ d(u_i - u_j, \hat{u}_i) - (-1)^i \int_S \lambda \hat{u}_i = 0 \quad \forall \hat{u}_i \in V_i \quad i, j = 1, 2, \ j \neq i$$

$$\frac{\epsilon}{\delta t}\int_S(\lambda^{n+1}-\lambda^n)\hat{\lambda}+\eta\int_S\frac{d\lambda}{ds}\frac{d\hat{\lambda}}{ds}=\int_S(u_1-u_2)\hat{\lambda}\quad\forall\hat{\lambda}\in L_h \tag{23}$$

where V_i and L_h are the finite element spaces of piecewise linear constinuous functions on Ω_i and S respectively, and

$$d(u,v)=\int_S(\alpha uv+\beta\frac{\partial u}{\partial s}\frac{\partial v}{\partial s})\quad S=\overline{\Omega}_1\cap\overline{\Omega}_2$$

The parameters chosen are:

$$\mu=10,\ \nu_2=0.1,\ \delta t=0.1,\ T=0.6,\ \alpha=1,\ \beta=0,\ \epsilon=1,\ \eta=0.1$$

with

$$\nu_2/\nu_1=10 \text{ in Test I and } \nu_2/\nu_1=100 \text{ in Test II.}$$

The results are shown on figure 1. The mesh has some 1500 vertices (2/3 in the top region Ω_1). Sensitivity with respect to numerical coefficients is mild except for ϵ and α. The size of λ is proportional to $1/\epsilon$ and its smoothness (or localization) is controlled by η; a good choice of α improves the quality of the results but β does not seem to play a major role. If ϵ is too small the method is unstable.

The numerical tests show that the method is feasible and well adapted to discontinuous coefficients. Different time steps in different sub-domains (with rendez-vous) improve the computing time and do not affect the precision.

References

[1] LIONS J.L., PIRONNEAU O., Algorithmes parallèles pour la solution de problèmes aux limites, C.R.A.S., 327, pp 947-352, Paris 1998.

[2] LIONS J.L., PIRONNEAU O., Sur le contrôle des sytèmes distribués. C.R.A.S., 327, pp 993-998, Paris 1998.

[3] LIONS J.L., PIRONNEAU O., Domain decomposition methods for CAD. C.R.A.S., 328, pp 73-80, Paris 1999.

[4] LIONS J.L., PIRONNEAU O., Overlapping Domain Decomposition of Evolution Operators C.R.A.S. à paraître.

[5] PIRONNEAU O., *Finite Element Methods for Fluid Flows.* J. Wiley, Chichester, 1987.

[6] ACHDOU Y., KUZNETSOV Y., PIRONNEAU O., Substructuring preconditioners for the Q_1 Mortar element method. *Numericsche Mathematik*, pages 419–449, 1995.

[7] LIONS, J.L., MAGENES, E. Problèmes aux limites non-homogènes et applications, Vol 1, Dunod 1968.

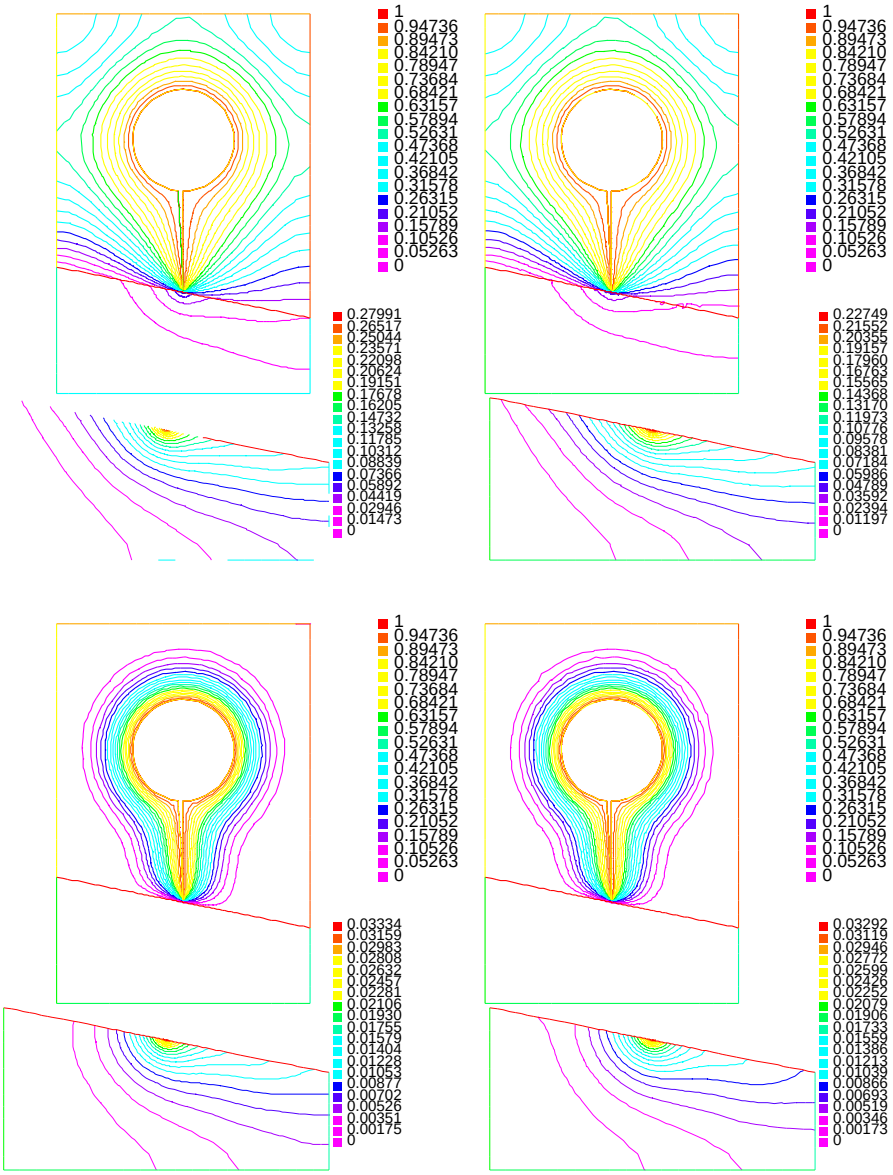

Figure 1: *Top 4 figures: Upper left: Solution I computed without decomposition. Upper right: Solution computed with decomposition. Below is a zoom of both on subdomain Ω_2 where the convection and diffusion are large. Bottom 4 figures: same for Test II where the ratio of diffusion coefficient is 100 instead of 10.*

8 Appendix: Stability of Algorithm(I)

Without loss of generality we can assume $f = 0$. Let us assume that a, b, c are symmetric; recall that for any bilinear form $e(,)$

$$2e(v^{n+1} - v^n, v^{n+1}) = e(v^{n+1}) - e(v^n) + e(v^{n+1} - v^n) \tag{24}$$
$$2e(v^{n+1} - v^n, v^n) = e(v^{n+1}) - e(v^n) - e(v^{n+1} - v^n) \tag{25}$$

Notations

$$e(v \equiv e(v, v) \quad \tilde{\epsilon}(\lambda, \mu) \equiv \frac{1}{\delta t}\epsilon(\lambda, \mu)_S \quad \tilde{c}(u, v) = \frac{1}{\delta t}c(u, v) \quad \delta u = (u_1 - u_2)_S.$$

Finally the indices i will stand for a sum for $i = 1, 2$; for instance $c_i(u_i)$ will stand for $c_1(u_1) + c_2(u_2)$.

Lemma 1

$$\begin{aligned}&\tilde{\epsilon}(\lambda^{n+1}) + \tilde{c}_i(u_i^{n+1}) + 2\eta b(\lambda^{n+1}) + \tilde{\epsilon}(\lambda^{n+1} - \lambda^n) + 2a_i(u_i^{n+1}) \\ &= \tilde{\epsilon}(\lambda^n) + \tilde{c}_i(u_i^n) + \tilde{c}_i(u_i^{n+1} - u_i^n) + 2a_i(u_i^{n+1} - u_i^n, u_i^{n+1})\end{aligned} \tag{26}$$

Proof If $(12)_a$ is used with $\hat{\lambda} = \lambda^{n+1}$ and $(12)_{b,c}$ is used with $\hat{u} = u_i^{n+1}$ then

$$\begin{aligned}&2\tilde{\epsilon}(\lambda^{n+1} - \lambda^n, \lambda^{n+1}) + 2\eta b(\lambda^{n+1}) - 2(\delta u^n, \lambda^{n+1})_S \\ &+2\tilde{c}_i(u_i^{n+1} - u_i^n) + 2a_i(u_i^{n+1}, u_i^n) + 2(\delta u^n, \lambda^{n+1})|S = 0\end{aligned}$$

Next we make use of (24) on $\tilde{\epsilon}$ and of (25) on $\tilde{c}_i$ and finally we note that $a_i(u_i^{n+1}, u_i^n) = a_i(u_i^{n+1}) - a_i(u_i^{n+1}, u_i^{n+1} - u_i^n)$.

Lemma 2

$$0 = \tilde{c}_i(u_i^{n+1} - u_i^n) + a_i(u_i^{n+1}, u_i^{n+1} - u_i^n) + (\lambda^{n+1}, \delta u^{n+1} - \delta u^n)_S \tag{27}$$

Proof
Choose $\hat{u}_i = u_i^{n+1} - u_i^n$ in $(12)_{b,c}$.

Lemma 3

$$\begin{aligned}&2\tilde{\epsilon}(\lambda^{n+1} - \lambda^n) + \eta b(\lambda^{n+1}) + \eta b(\lambda^{n+1} - \lambda^n) \\ &= \eta b(\lambda^n) + 2(\delta u^n, \lambda^{n+1} - \lambda^n)_S\end{aligned} \tag{28}$$

Proof
Choose $\hat{\lambda} = \lambda^{n+1} - \lambda^n$ in $(12)_a$ multiplied by 2, and use (24) on $b(\lambda^{n+1}, \lambda^{n+1} - \lambda^n)$.

Lemma 4

$$\begin{aligned}&2\tilde{\epsilon}(\lambda^{n+1}) + 2c_i(u_i^{n+1}) + a_i(u_i^{n+1}) + \eta b(\lambda^{n+1}) + \eta b(\lambda^{n+1} + \lambda^n) \\ &+2(\lambda^{n+1}, \delta u^{n+1})_S \\ &\leq 2\tilde{\epsilon}(\lambda^n) + 2\tilde{c}_i(u_i^n) + a_i(u_i^n) + 2(\lambda^n, \delta u^n)_S + \eta b(\lambda^n)\end{aligned} \tag{29}$$

Proof
First add twice (27) to (28); it yields

$$\begin{aligned}&\eta b(\lambda^{n+1}) + 2\tilde{\epsilon}(\lambda^{n+1} - \lambda^n) + \eta b(\lambda^{n+1} - \lambda^n)\\ &-2(\lambda^{n+1}, \delta u^{n+1})_S\\ &= \eta b(\lambda^n) - 2(\lambda^n, \delta u^n)_S + 2\tilde{c}(u_i^{n+1} - u_i^n) + 2a_i(u_i^{n+1}, u_i^{n+1} - u_i^n) \quad (30)\end{aligned}$$

Now by subtracting (30) to twice (26) we find that

$$\begin{aligned}&2\tilde{\epsilon}(\lambda^{n+1}) + 2c_i(u_i^{n+1}) + 3\eta b(\lambda^{n+1}) - \eta b(\lambda^{n+1} - \lambda^n) + 4a_i(u_i^{n+1})\\ &+2(\lambda^{n+1}, \delta u^{n+1})_S\\ &= 2\tilde{\epsilon}(\lambda^n) + 2\tilde{c}_i(u_i^n) + 2a_i(u_i^{n+1} - u_i^n, u_i^{n+1}) - \eta b(\lambda^n) + 2(\lambda^n, \delta u^n)_S\end{aligned}$$

Finally we make use of the following inequalities

$$\begin{aligned}&-2a_i(u_i^n, u_i^{n+1}) \leq a_i(u_i^n) + a_i(u_i^{n+1})\\ &3b(\lambda^{n+1}) - b(\lambda^{n+1} - \lambda^n) + b(\lambda^n) = b(\lambda^{n+1}) + b(\lambda^{n+1} + \lambda^n) - b(\lambda^n)\end{aligned}$$

Proposition 1 *Provided that $\epsilon\delta t + \eta/2 \geq 1$, Algorithm(I) is stable if*

$$\frac{2}{\delta t}c_j(u_j) + a_j(u_j) \geq \int_S u_j^2 \quad \forall u_j \in V_j, \quad j = 1, 2.$$

Proof
When (29) is summed up from $n = 0$ to $n = N$, most terms cancel. The only problematic term left is $2(\lambda^{N+1}, \delta u_i^{N+1})_S$. It is bounded by

$$2(\lambda^{N+1}, \delta u^{N+1})_S \leq \int_S [2\lambda^{N+1^2} + u_1^{N+1^2} + u_2^{N+1^2}]$$

So we are left with

$$2\tilde{\epsilon}(\lambda^{N+1}) + 2\tilde{c}_i(u_i^{N+1}) + a_i(u_i^{N+1}) + \eta b(\lambda^{N+1}) \leq C + 2|\lambda^{N+1}|_S^2 + |u_i^{N+1}|_S^2$$

Remark 4 *Note that in the discrete case the condition is satisified if either*

$$\frac{2\gamma_0}{\delta t} \geq \frac{C}{h}$$

where C is the discrete inverse inequality constant:

$$C = h \max_{i=1,2,u_i \in V_i} \frac{\int_S u_i^2}{\int_{\Omega_i} u_i^2}$$

or if $\alpha > c$ where c is the trace-theorem constant

$$c = \max_{i=1,2,u_i \in H^1(\Omega_i)} \frac{\int_{\Omega_i} (u_i^2 + |\nabla u_i|^2)}{\int_S u_i^2}$$