ELSEVIER

Comput. Methods Appl. Mech. Engrg. 184 (2000) 145–170

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

A domain decomposition preconditioner for an advection–diffusion problem

Yves Achdou [a,*], Patrick Le Tallec [b], Frédéric Nataf [c], Marina Vidrascu [d]

[a] *INSA Rennes, 20 Av des Buttes de Coesmes, 35043 Rennes, France*
[b] *Université Paris Dauphine, 75 775 Paris Cedex 16, France*
[c] *CMAP UMR7641 CNRS, Ecole Polytechnique, 91128, Palaiseau Cedex, France*
[d] *INRIA Rocquencourt, BP 105, 78 153 Le Chesnay Cedex, France*

Received 1 February 1999

Abstract

We propose a generalization of the Neumann–Neumann algorithm for advection–diffusion problems: the Neumann conditions are replaced by suitable Robin conditions. The method is first introduced and analysed in the case where the domain is partitioned into vertical strips, then generalized to an arbitrary decomposition. A coarse space procedure is proposed. Finally, numerical results for bidimensional and tridimensional tests are given. © 2000 Elsevier Science S.A. All rights reserved.

1. Introduction

The primary objective of domain decomposition methods is the efficient solution on parallel machines of problems in Computational Mechanics set on complex geometries and approximated on very fine grids. Most methods refer to iterative substructuring techniques (the so-called Schur domain decomposition method) and use non-overlapping domain partitions which split the original domain into small disjoint subdomains and reduce the problem to an interface problem solved by an iterative method. The main issue conditioning the success and parallel efficiency of such techniques is the choice of the preconditioner which must have nice parallel properties, must be able to handle arbitrary elliptic operators and discretization grids, and its performance must be insensitive to the discretization step h and to the number of subdomains. Many such preconditioners have appeared in the literature, following the early work of Bramble, Pasciak and Schatz [4] ([8,9,11,15,24]). For three-dimensional elasticity, efficient results have been obtained using either the wire-basket algorithm of Smith [21] or the Neumann–Neumann preconditioner of [23,22]. For advection–diffusion or Helmholtz problems, one can also choose artificial boundary conditions as interface conditions to improve the condition number [10,12,13,19]. For the analysis of overlapping domain decomposition methods for non-symmetric problems, see [5–7].

We propose herein a generalization of the Neumann–Neumann preconditioner for the Schur domain decomposition method applied to an advection–diffusion equation, by replacing the local Neumann boundary conditions by suitable Robin boundary conditions which take into account the non-symmetry of the operator. The choice of these conditions comes from a Fourier analysis, which is given in Section 2. When the operator is symmetric the proposed Robin boundary conditions reduce to Neumann boundary

[*] Corresponding author.

0045-7825/00/$ - see front matter © 2000 Elsevier Science S.A. All rights reserved.
PII: S 0 0 4 5 - 7 8 2 5 (9 9) 0 0 2 2 7 - 3

conditions. Also, as in the symmetric case, the proposed preconditioner is exact for two subdomains and a uniform velocity.

The paper is organized as follows. In Section 2, the method is defined and analyzed for a decomposition into strips at the continuous level. In particular, it is shown that the preconditioned system is close to an idempotent operator. Then, the proposed preconditioner is constructed directly at the algebraic level. Next, a variational writing is introduced in Section 3 which makes it possible to generalize the proposed approach to general partitions and to include a coarse space. Numerical results illustrating the performances of the proposed technique conclude the paper.

2. The vertical strip case

The general case of an arbitrary domain decomposition is treated in Section 3.

2.1. The continuous case – definition of the algorithm

We consider a rectangular domain $\Omega =]0, H[\times] - \eta, \eta[$, and the advection–diffusion boundary value problem in Ω

$$\begin{aligned} \mathscr{L}(u) \equiv cu + \vec{a} \cdot \nabla u - \nu \Delta u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{1}$$

The positive constant c may arise from a time discretization by an Euler implicit scheme of the time dependent equation. The equation is solved by a primal Schur method. We focus on the case when the domain is decomposed into non-overlapping vertical strips $\Omega_k = (l_k, l_{k+1}) \times] - \eta, \eta[, 1 \leqslant k \leqslant N$. Let $\Gamma_{k,k+1} = \{l_{k+1}\} \times] - \eta, \eta[$. The unit outward normal vector to domain Ω_k is denoted by $\vec{n}_k$.

We introduce

$$\begin{aligned} \mathscr{S} : \left(H_{00}^{1/2}(] - \eta, \eta[)\right)^{N-1} \times L^2(\Omega) &\to (H^{-1/2}(] - \eta, \eta[))^{N-1} \\ ((u_k)_{1 \leqslant k \leqslant N-1}, f) &\mapsto \left(\frac{1}{2} \nu \left(\frac{\partial v_k}{\partial n_k} + \frac{\partial v_{k+1}}{\partial n_{k+1}} \right)_{\Gamma_{k,k+1}} \right)_{1 \leqslant k \leqslant N-1}, \end{aligned} \tag{2}$$

where v_k satisfies

$$\begin{aligned} &\mathscr{L}(v_k) = f \quad \text{in } \Omega_k, \\ &\quad v_k = u_k \quad \text{on } \Gamma_{k,k+1} \text{ for } 1 \leqslant k \leqslant N-1, \\ &\quad v_k = u_{k-1} \quad \text{on } \Gamma_{k-1,k} \text{ for } 2 \leqslant k \leqslant N, \\ &\quad v_k = 0 \quad \text{on } \partial\Omega \cap \partial\Omega_k. \end{aligned} \tag{3}$$

It is clear that $U = (u_{\Gamma_{k,k+1}})_{1 \leqslant k \leqslant N-1}$ satisfies

$$\mathscr{S}(U, 0) = -\mathscr{S}(0, f).$$

At the continuous level, we propose an approximate inverse of $\mathscr{S}(., 0)$ defined by

$$\begin{aligned} \mathscr{T} : (H^{-1/2}(] - \eta, \eta[))^{N-1} &\to (H_{00}^{1/2}(] - \eta, \eta[))^{N-1}, \\ (g_k)_{1 \leqslant k \leqslant N-1} &\mapsto \left(\frac{1}{2} (v_k + v_{k+1})_{\Gamma_{k,k+1}} \right)_{1 \leqslant k \leqslant N-1}, \end{aligned} \tag{4}$$

where v_k satisfies

$$
\begin{aligned}
&\mathscr{L}(v_k) = 0 \quad \text{in } \Omega_k,\\
&\left(\nu\frac{\partial}{\partial n_k} - \frac{\vec{a}\cdot\vec{n}_k}{2}\right)(v_k) = g_k \quad \text{on } \Gamma_{k,k+1} \text{ for } 1 \leqslant k \leqslant N-1,\\
&\left(\nu\frac{\partial}{\partial n_k} - \frac{\vec{a}\cdot\vec{n}_k}{2}\right)(v_k) = g_{k-1} \quad \text{on } \Gamma_{k-1,k} \text{ for } 2 \leqslant k \leqslant N,\\
&v_k = 0 \quad \text{on } \partial\Omega \cap \partial\Omega_k.
\end{aligned}
\tag{5}
$$

Remark 1. *In Section* 3, *the boundary value problem* (5) *is proved to be well-posed* (*see also* [1]) *under the usual condition* $c - 1/2 \operatorname{div} \vec{a} > 0$.

Remark 2. *Our approach is different from that used in* [19] *or* [12], *where the interface conditions* $\nu(\partial/\partial n_k) - \min(\vec{a}\cdot\vec{n}_k, 0)$ *are used in the framework of Schwarz algorithms.*

In Section 2.2, we analyze the preconditioner in the special case where $\eta = \infty$, for simplicity.

2.2. Theoretical study of the algorithm – uniform velocity

Let us first recall the

Proposition 2.1 (see [1]). *In the case where the plane* $\mathbb{R}^2$ *is decomposed into the left* ($\Omega_1 =]-\infty, 0[\times\mathbb{R}$) *and right* ($\Omega_2 =]0, \infty[\times\mathbb{R}$) *half-planes, and where the velocity* $\vec{a}$ *is uniform, we have that*

$$\mathscr{T} \circ \mathscr{S}(.,0) = Id.$$

This result shows that, as it is the case for the Neumann–Neumann algorithm applied to symmetric operators, the Robin–Robin preconditioner is exact for a two domain decomposition.

For a more general decomposition into an arbitrary number of vertical strips, the vector $\vec{a}$ is denoted $\vec{a} = (a_x, a_y)$. We assume without loss of generality that the component of the velocity normal to the interface is positive $a_x \geqslant 0$. Let H denote the minimum of the widths of the subdomains. The main result of the section is Theorem 3 below: it states that for $\exp\{-(a_x + \sqrt{a_x^2 + 4\nu c})H/\nu\}$ small enough, the preconditioned system is close to an idempotent operator of order $[N/2]$, where $[x]$ denotes the integer part of x. This has an important effect on the convergence of GMRES applied to the preconditioned system, as it will be seen from Corollary 6 and Fig. 2.

For Theorem 3, we introduce $\mathscr{H}_N^s = \prod_{k=1}^{N-1} H^s(\Gamma_{k,k+1})$, $s \in \mathbb{R}$ the space of H^s functions on the $N-1$ interfaces. Let $U \in \mathscr{H}_N^s$ be denoted by $U = (u_k)_{1\leqslant k\leqslant N-1}$. The space $\mathscr{H}_N^s$ is endowed with the norm $\|U\|_{\mathscr{H}_{N,\infty}^s} = \sup_{1\leqslant k\leqslant N-1} \|u_k\|_{H^s(\Gamma_k,k+1)}$.

Theorem 3. *Assume that the velocity field* $\vec{a}$ *is uniform. Let* $\Omega = (0, H_\Omega) \times \mathbb{R}$ *be decomposed into* N *non-overlapping vertical strips* $\Omega_k = (l_k, l_{k+1}) \times \mathbb{R}$, $0 \leqslant k \leqslant N-1$ *and* $\Gamma_{k,k+1} = \{l_{k+1}\} \times \mathbb{R}$.

Let H *be the size of the smallest subdomain,* $H = \min_k(l_{k+1} - l_k)$. *Assume that*

$$\epsilon \equiv \exp\{-(a_x + \sqrt{a_x^2 + 4\nu c})H/\nu\} < 1, \tag{6}$$

and that

$$\rho \equiv 3N\epsilon/(1-\epsilon)^{N+1} < 1.$$

Then, for $n \geqslant [N/2]$, *we have*

$$\|(\mathscr{T} \circ \mathscr{S}(.,0) - Id)^n\|_{L(\mathscr{H}_N^s)} \leqslant \frac{N}{2(1-\rho)} \frac{1}{(1-\epsilon)^{N-2}} \rho^{[n/[N/2]]-1}, \tag{7}$$

where $[x]$ *denotes the integer part of* x.

Remark 4. *The above result explains why the GMRES algorithm stagnates initially during the* [N/2] *first iterations and then converges fast*, *see Corollary* 6 *and Figs.* 2 *and* 3. *This behavior is classical for convection–diffusion type problems*, *see for instance* [19] *or* [13].

Remark 5. *In fact*, *as it can be seen from formulas* (13) *below*, *as soon as*

$$\exp\{(a_x - \sqrt{a_x^2 + 4\nu c})H/\nu\} \ll 1,$$

the operator $\mathscr{T} \circ \mathscr{S}$ *is already very close to identity and a better estimate can easily be obtained*:

$$\|(\mathscr{T} \circ \mathscr{S}(.,0) - Id)\|_{L(\mathscr{H}_N^s)} \leqslant \frac{6\exp\{(a_x - \sqrt{a_x^2 + 4\nu c})H/\nu\}}{(1-\epsilon)^3}. \tag{8}$$

The above theorem is thus interesting when $\mathrm{e}^{(a_x - \sqrt{a_x^2+4\nu c})H/\nu}$ *is close to* 1. *This corresponds for instance to the case of a very large time step with a strong laminar flow*.

On the other hand, *the above results cannot be applied when the operator is symmetric and very close to a Laplace operator* ($c \ll 1$). *Indeed*, *in order to have good convergence rates*, *it is then necessary to modify the Robin–Robin preconditioner* (*which amounts to the Neumann–Neumann preconditioner*) *by adding a coarse space and a pseudo inverse for the Neumann problem* (5).

Proof. The width of subdomain Ω_k is denoted by $H_k, H_k = l_{k+1} - l_k$. The proof is based on the Fourier transform in the y direction and the Fourier variable is denoted by ξ. The inverse Fourier transform is denoted by $\mathscr{F}^{-1}$. The action of $\mathscr{S}(.,0)$ can be expressed in terms of Fourier transform

$$\mathscr{S}(.,0)(U) = \mathscr{F}^{-1}(\widehat{\mathscr{S}}(\xi)\hat{U}(\xi)), \quad U \in \mathscr{H}_N^s,$$

where $\widehat{\mathscr{S}}$ is a tridiagonal $(N-1) \times (N-1)$ symbol-valued matrix. For instance, the value of $\widehat{\mathscr{S}}_{k+1,k+1+j}, j = -1, 0, 1$ is computed in the following way. Let $u_m \in H^s(\Gamma_{m,m+1})$, $m = k-1, k, k+1$. Let v_k satisfy

$$\mathscr{L}(v_k) = 0 \quad \text{in } \Omega_k, \tag{9}$$

with the boundary conditions

$$\begin{aligned} v_k &= u_k \quad \text{on } \Gamma_{k-1,k}, \\ v_k &= u_{k+1} \quad \text{on } \Gamma_{k,k+1}, \end{aligned} \tag{10}$$

and

$$\begin{aligned} \mathscr{L}(v_{k+1}) &= 0 \quad \text{in } \Omega_{k+1}, \\ v_{k+1} &= u_{k+1} \quad \text{on } \Gamma_{k,k+1}, \\ v_{k+1} &= u_{k+2} \quad \text{on } \Gamma_{k+1,k+2}. \end{aligned}$$

We shall compute the Fourier transform of $1/2\nu((\partial v_k/\partial n_k) + (\partial v_{k+1}/\partial n_{k+1}))\Gamma_{k,k+1}$. The Fourier transform of (9) w.r.t. y yields

$$(c + a_x\partial_x + a_y i\xi - \nu\partial_{xx} + \nu\xi^2)(\hat{v}_k(x,\xi)) = 0,$$

where $i^2 = -1$. For a given ξ, this equation is an ordinary differential equation in x whose solutions have the form $\alpha(\xi)\exp\{\lambda^-(\xi)(x - l_k)\} + \beta(\xi)\exp\{\lambda^+(\xi)(x - l_{k+1})\}$, where

$$\lambda^{\pm}(\xi) = \frac{a_x \pm \sqrt{4\nu c + a_x^2 + 4ia_y\xi\nu + 4\xi^2\nu^2}}{2\nu}. \tag{11}$$

We have, $\mathrm{Re}(\lambda^{\pm}(\xi)) \gtrless 0$, where $\mathrm{Re}(z)$ denotes the real part of a complex number z. The Dirichlet boundary conditions (10) yield the values of α and β:

$$\begin{aligned}\alpha(\xi) &= \frac{\hat{u}_k(\xi) - \mathrm{e}^{-\lambda^+(\xi)H_k}\hat{u}_{k+1}(\xi)}{1-\mathrm{e}^{(\lambda^-(\xi)-\lambda^+(\xi))H_k}},\\ \beta(\xi) &= \frac{\hat{u}_{k+1}(\xi) - \mathrm{e}^{\lambda^-(\xi)H_k}\hat{u}_k(\xi)}{1-\mathrm{e}^{(\lambda^-(\xi)-\lambda^+(\xi))H_k}}.\end{aligned}$$

Hence,

$$\begin{aligned}\frac{\partial\hat{v}_k}{\partial n_k}\Gamma_{k,k+1} &= \frac{\partial\hat{v}_k}{\partial x}\Gamma_{k,k+1}\\ &= \hat{u}_k\frac{v(\lambda^- - \lambda^+)\mathrm{e}^{\lambda^- H_k}}{1-\mathrm{e}^{(\lambda^- -\lambda^+)H_k}} + \hat{u}_{k+1}\frac{v\lambda^+ - v\lambda^-\mathrm{e}^{(\lambda^- -\lambda^+)H_k}}{1-\mathrm{e}^{(\lambda^- -\lambda^+)H_k}}.\end{aligned}$$

The expression for $(\partial\hat{v}_{k+1})/(\partial n_{k+1}\Gamma_{k,k+1})$ can be obtained in the same manner. Then, we obtain

$$\begin{aligned}\frac{v}{2}\left(\frac{\partial\hat{v}_k}{\partial n_k} + \frac{\partial\hat{v}_{k+1}}{\partial n_{k+1}}\right)_{\Gamma_{k,k+1}} &= \hat{u}_k\frac{v(\lambda^- - \lambda^+)\mathrm{e}^{\lambda^- H_k}}{2(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_k})} + \hat{u}_{k+1}\left[\frac{v\lambda^+ - v\lambda^-\mathrm{e}^{(\lambda^- -\lambda^+)H_k}}{2(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_k})} + \frac{-v\lambda^- + v\lambda^+\mathrm{e}^{(\lambda^- +\lambda^+)H_{k+1}}}{2(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}})}\right]\\ &\quad + \hat{u}_{k+2}\frac{v(\lambda^- - \lambda^+)\mathrm{e}^{-\lambda^+ H_{k+1}}}{2(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}})}.\end{aligned}$$

This yields the following formulas for $\widehat{\mathscr{S}}$

$$\begin{aligned}\widehat{\mathscr{S}}_{k+1,k} &= \frac{v(\lambda^- - \lambda^+)\mathrm{e}^{\lambda^- H_k}}{2(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_k})},\\ \widehat{\mathscr{S}}_{k+1,k+1} &= \frac{v\lambda^+ - v\lambda^-\mathrm{e}^{(\lambda^- -\lambda^+)H_k}}{2(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_k})} + \frac{-v\lambda^- + v\lambda^+\mathrm{e}^{(\lambda^- +\lambda^+)H_{k+1}}}{2(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}})},\\ \widehat{\mathscr{S}}_{k+1,k+2} &= \frac{v(\lambda^- - \lambda^+)\mathrm{e}^{-\lambda^+ H_{k+1}}}{2(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}})}.\end{aligned} \tag{12}$$

Similarly, we can compute the symbol-valued matrix $\widehat{\mathscr{T}}$:

$$\widehat{\mathscr{T}}(G) = \mathscr{F}^{-1}(\widehat{\mathscr{T}}(\xi)\hat{G}(\xi)), \quad G \in \mathscr{H}^s_N.$$

Hence, the preconditioned system reads

$$\mathscr{T}\mathscr{S}(.,0)(U) = \mathscr{F}^{-1}(\widehat{\mathscr{T}}\widehat{\mathscr{S}}(\xi)\hat{U}(\xi)), \quad U \in \mathscr{H}^s_N,$$

where $\widehat{\mathscr{T}}\widehat{\mathscr{S}}$ is a pentadiagonal matrix. After a tedious computation, we obtain that its non-zero entries are given by the following formulas:

$$\begin{aligned}(\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{k,k+2} &= -\frac{\mathrm{e}^{-\lambda^+(H_{k+1}+H_{k+2})}}{(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+2}})(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}})},\\ (\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{k,k+1} &= \frac{\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+2}} - \mathrm{e}^{(\lambda^- -\lambda^+)H_k}}{(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_k})(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}})(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+2}})}\mathrm{e}^{-\lambda^+ H_{k+1}},\\ (\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{k,k} &= 1 + \frac{\mathrm{e}^{(\lambda^- -\lambda^+)H_k} + \mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}}}{(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_k})(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}})},\\ (\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{k,k-1} &= \frac{\mathrm{e}^{(\lambda^- -\lambda^+)H_{k-1}} - \mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}}}{(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k-1}})(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_k})(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k+1}})}\mathrm{e}^{\lambda^- H_k},\\ (\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{k,k-2} &= -\frac{\mathrm{e}^{\lambda^-(H_{k-1}+H_{k-2})}}{(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k-2}})(1-\mathrm{e}^{(\lambda^- -\lambda^+)H_{k-1}})}.\end{aligned} \tag{13}$$

The above expressions are valid for any k as long as the indices make sense and except for the following entries of the matrix $\widehat{\mathscr{T}}\widehat{\mathscr{S}}$, where the Dirichlet boundary conditions (5) in the Robin–Robin preconditioner yield some minor modifications:

$$\begin{aligned}
(\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{1,1} &= 1 + \frac{\mathrm{e}^{2(\lambda^- - \lambda^+)H_1} + \mathrm{e}^{(\lambda^- - \lambda^+)H_2}}{(1 - \mathrm{e}^{2(\lambda^- - \lambda^+)H_1})(1 - \mathrm{e}^{(\lambda^- - \lambda^+)H_2})},\\
(\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{1,2} &= \frac{\mathrm{e}^{(\lambda^- - \lambda^+)H_3} + \mathrm{e}^{(\lambda^- - \lambda^+)H_1}}{(1 - \mathrm{e}^{(\lambda^- - \lambda^+)H_3})(1 - \mathrm{e}^{(\lambda^- - \lambda^+)H_2})(1 + \mathrm{e}^{(\lambda^- - \lambda^+)H_1})}\mathrm{e}^{-\lambda^+ H_2},\\
(\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{N-1,N-1} &= 1 + \frac{\mathrm{e}^{2(\lambda^- - \lambda^+)H_N} + \mathrm{e}^{(\lambda^- - \lambda^+)H_{N-1}}}{(1 - \mathrm{e}^{2(\lambda^- - \lambda^+)H_N})(1 - \mathrm{e}^{(\lambda^- - \lambda^+)H_{N-1}})},\\
(\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{N-1,N-2} &= \frac{(\mathrm{e}^{(\lambda^- - \lambda^+)H_{N-2}} + \mathrm{e}^{(\lambda^- - \lambda^+)H_N})\mathrm{e}^{\lambda^- H_{N-1}}}{(1 - \mathrm{e}^{(\lambda^- - \lambda^+)H_{N-2}})(1 - \mathrm{e}^{(\lambda^- - \lambda^+)H_{N-1}})(1 + \mathrm{e}^{(\lambda^- - \lambda^+)H_N})}.
\end{aligned}$$

For large values of ξ, it is clear from the expressions of $\lambda^{\pm}(\xi)$ that $\widehat{\mathscr{T}}\widehat{\mathscr{S}}$ is very close to the identity. But, for ξ close to zero and $c \ll 1$ (quasi steady state problems), the extra diagonal entries $(\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{k,k-2}$ are close to -1 while the other extra-diagonal entries remain small. This is the reason why a theoretical study of the convergence has to be done. For this, matrix $\widehat{\mathscr{T}}\widehat{\mathscr{S}}$ is written as the sum of three matrices: the Identity, a nilpotent matrix of order $[N/2]$ ($\widehat{\mathscr{A}}$) and a small perturbation ($\widehat{\mathscr{B}}$):

$$\widehat{\mathscr{T}}\widehat{\mathscr{S}} = Id + \widehat{\mathscr{A}} + \widehat{\mathscr{B}}, \tag{14}$$

where

$$\widehat{\mathscr{A}} = (\widehat{\mathscr{A}}_{k,l})_{1 \leqslant k,l \leqslant N-1} = \begin{cases} (\widehat{\mathscr{T}}\widehat{\mathscr{S}})_{k,l} & \text{for } l = k-2,\ 3 \leqslant k \leqslant N-1,\\ 0 & \text{otherwise.}\end{cases}$$

The value of $\widehat{\mathscr{B}}$ is given by (14). From the structure of $\widehat{\mathscr{A}}$, it is clear that

$$\widehat{\mathscr{A}}^{[N/2]} = 0.$$

In order to estimate the norms of the matrices, we first have to define the norms that will be used. Let $x = (x_1, \ldots, x_{N-1})$ be a vector in $\mathbb{R}^{N-1}$ and $\|x\|_\infty$ denote the maximum norm, $\|x\|_\infty = \sup_{1 \leqslant k \leqslant N-1} |x_k|$. For a matrix $D = (D_{ij})_{1 \leqslant i,j \leqslant N-1} \in \mathbb{R}^{(N-1)\times(N-1)}$, the corresponding operator norm is $\|D\|_\infty = \sup_i \Sigma_j \mid D_{ij} \mid$: $\|D_x\|_\infty \leqslant \|D\|_\infty \|x\|_\infty$.

By applying the result of [17] (see also [18]), we have the following estimate.

Proposition 2.2. *If*

$$\rho(\xi) \equiv \|\widehat{\mathscr{B}}(\xi)\|_\infty \sum_{i=0}^{[N/2]-1} \|\widehat{\mathscr{A}}(\xi)\|_\infty^i < 1. \tag{15}$$

Then, define $C(\xi)$ *by*

$$C(\xi) \equiv \frac{1}{1 - \rho(\xi)} \sum_{i=0}^{[N/2]-1} \|\widehat{\mathscr{A}}(\xi)\|_\infty^i \tag{16}$$

we have for $n \geqslant [N/2]$,

$$\|(\widehat{\mathscr{A}} + \widehat{\mathscr{B}})^n(\xi)\|_\infty = \|(\widehat{\mathscr{T}}\widehat{\mathscr{S}} - Id)^n(\xi)\|_\infty \leqslant C(\xi)\rho(\xi)^{[n/[N/2]]-1}. \tag{17}$$

Going back to the proof of Theorem 3, we shall estimate $\rho(\xi)$ independently of ξ. Let ϵ be as in Theorem 3,

$$|\mathrm{e}^{-\lambda^+(\xi)H_k}| \leqslant \epsilon$$

for any ξ. Therefore, the norm of $\widehat{\mathscr{A}}(\xi)$ is bounded by $1/(1-\epsilon)^2$ and that of $\widehat{\mathscr{B}}(\xi)$ by $5\epsilon/(1-\epsilon)^3$. Then, by estimating $\|\widehat{\mathscr{A}}(\xi)\|_\infty^i$ by $1/(1-\epsilon)^{N-2}$ for $i \leqslant [N/2]$ we get

$$\sum_{i=0}^{[N/2]-1} \|\widehat{\mathscr{A}}\|_\infty^i \leqslant \frac{N}{2} \frac{1}{(1-\epsilon)^{N-2}}.$$

Hence, $\rho(\xi)$ may be roughly estimated by $3N\epsilon/(1-\epsilon)^{N+1}$. By using this estimate and Parseval–Plancherel theorem, we can conclude.

We now consider the implications of Theorem 3 for the GMRES algorithm (see [20]) applied to the problem

$$(\mathscr{T} \circ \mathscr{S}(.,0))(U) = -\mathscr{T} \circ \mathscr{S}(0,f). \tag{18}$$

Since $\mathscr{T} \circ \mathscr{S}(.,0)$ is a bounded operator from $\mathscr{H}_N^s$ into itself, it is possible to define a "continuous" GMRES algorithm. For this, we use a norm induced by a scalar product.

The space $\mathscr{H}_N^s$ is endowed with the norm

$$\|U\|_{\mathscr{H}_{N,2}^s} = \sqrt{\sum_{k=1}^{N-1} \|u_k\|^2_{H^s(\Gamma_{k,k+1})}}.$$

We have

$$\|U\|_{\mathscr{H}_{N,\infty}^s} \leqslant \|U\|_{\mathscr{H}_{N,2}^s} \leqslant \sqrt{N-1}\,\|U\|_{\mathscr{H}_{N,\infty}^s}.$$

For the sake of simplicity, the GMRES algorithm is considered with an initial guess, $U_{\mathrm{gmres}}^0 = 0$. Hence, the initial residual is

$$r^0 = \mathscr{T} \circ \mathscr{S}(0,f).$$

The Krylov space $\mathscr{K}^n(r^0)$ is given by

$$\mathscr{K}^n = \mathrm{Span}\{r^0, (\mathscr{T} \circ \mathscr{S}(.,0))(r^0), \ldots, (\mathscr{T} \circ \mathscr{S}(.,0))^{n-1}(r^0)\}.$$

The GMRES iterate U_{gmres}^n is defined by

$$\|(\mathscr{T} \circ \mathscr{S}(.,0))(U_{\mathrm{gmres}}^n) + r^0\|_{\mathscr{H}_{N,2}^s} = \min_{U \in \mathscr{K}^n} \|\mathscr{T} \circ \mathscr{S}(.,0))(U) + r^0\|. \tag{19}$$

Since the $\|.\|_{\mathscr{H}_{N,2}^s}$ norm is induced by a scalar product, the above minimization problem amounts to the solving of a $n \times n$ square linear system. We have the

Corollary 6. *Let $r^0 = \mathscr{T} \circ \mathscr{S}(0,f)$ be the initial residual. Let U_{gmres}^n be defined as in* (19) *and the residual at step n be defined by $r^n = (\mathscr{T} \circ \mathscr{S}(.,0))(U_{\mathrm{gmres}}^n) + r^0$. Then, under the assumptions of Theorem* 3, *we have the following estimate for $n \geqslant [N/2]$*

$$\|r^n\|_{\mathscr{H}_{N,2}^s} \leqslant C\rho^{n/[N/2]-1}\|r^0\|_{\mathscr{H}_{N,2}^s},$$

where

$$C = \sqrt{N-1}\left(1 + \frac{5\epsilon}{(1-\epsilon)^3} + \frac{1}{(1-\epsilon)^2}\right) \frac{N}{2(1-\rho)^2(1-\epsilon)^{N-2}}$$

and ρ and ϵ are defined in Theorem 3.

Remark 7. *Since this result is stated at the continuous level, it is a clear indication that the same kind of estimate should remain valid as the mesh size tends to zero.*

Proof. The proof is based on a comparison between a fixed point algorithm and the GMRES method defined above. Indeed, we introduce the following algorithm

$$\begin{aligned}&U^0_{\text{fixed pt}} = 0,\\ &U^{n+1}_{\text{fixed pt}} = (Id - \mathcal{T} \circ \mathcal{S}(.,0))\left(U^n_{\text{fixed pt}}\right) - \mathcal{T} \circ \mathcal{S}(0,f).\end{aligned}$$

It can easily be checked that $U^n_{\text{fixed pt}} \in \mathcal{K}^n$. Then, by (19), we have

$$\|(\mathcal{T} \circ \mathcal{S}(.,0))(U^n_{\text{gmres}}) + r^0\|_{\mathcal{H}^s_{N,2}} \leqslant \|(\mathcal{T} \circ \mathcal{S}(.,0))(U^n_{\text{fixed pt}}) + r^0\|_{\mathcal{H}^s_{N,2}}.$$

We now have to estimate the right-hand side of the above inequality. We have

$$U^n_{\text{fixed pt}} = -\sum_{i=0}^{n-1}(Id - \mathcal{T} \circ \mathcal{S}(.,0))^i \mathcal{T} \circ \mathcal{S}(0,f).$$

Under the assumptions of Theorem 3, this series converges to the solution U of (18),

$$U = -\sum_{i=0}^{\infty}(Id - \mathcal{T} \circ \mathcal{S}(.,0))^i \mathcal{T} \circ \mathcal{S}(0,f).$$

Hence,

$$\begin{aligned}(\mathcal{T} \circ \mathcal{S}(.,0))(U^n_{\text{fixed pt}}) + r^0 &= (\mathcal{T} \circ \mathcal{S}(.,0))(U^n_{\text{fixed pt}} - U)\\ &= -\mathcal{T} \circ \mathcal{S}(.,0)\Sigma^{\infty}_{i=n}(Id - \mathcal{T} \circ \mathcal{S}(.,0))^i \mathcal{T} \circ \mathcal{S}(0,f)\\ &= -\mathcal{T} \circ \mathcal{S}(.,0)\Sigma^{\infty}_{i=n}(Id - \mathcal{T} \circ \mathcal{S}(.,0))^i r^0.\end{aligned}$$

This gives

$$\begin{aligned}\|(\mathcal{T} \circ \mathcal{S}(.,0))(U^n_{\text{fixed pt}}) + r^0\|_{\mathcal{H}^s_{N,2}} &\leqslant \|\mathcal{T} \circ \mathcal{S}(.,0)\Sigma^{\infty}_{i=n}(Id - \mathcal{T} \circ \mathcal{S}(.,0))^i r^0\|_{\mathcal{H}^s_{N,2}}\\ &\leqslant \|\mathcal{T} \circ \mathcal{S}(.,0)\Sigma^{\infty}_{i=n}(Id - \mathcal{T} \circ \mathcal{S}(.,0))^i\|_{\mathcal{H}^s_{N,2}}\|r^0\|_{\mathcal{H}^s_{N,2}}\\ &\leqslant \sqrt{N-1}\|\mathcal{T} \circ \mathcal{S}(.,0)\Sigma^{\infty}_{i=n}(Id - \mathcal{T} \circ \mathcal{S}(.,0))^i\|_{\mathcal{H}^s_{N,\infty}}\|r^0\|_{\mathcal{H}^s_{N,2}}\\ &\leqslant \sqrt{N-1}\|\mathcal{T} \circ \mathcal{S}(.,0)\|_{\mathcal{H}^s_{N,\infty}}\Sigma^{\infty}_{i=n}\|(Id - \mathcal{T} \circ \mathcal{S}(.,0))^i\|_{\mathcal{H}^s_{N,\infty}}\|r^0\|_{\mathcal{H}^s_{N,2}}.\end{aligned}$$

The norm of $\mathcal{T} \circ \mathcal{S}(.,0)$ is estimated by $1 + 5\epsilon/(1-\epsilon)^3 + 1/(1-\epsilon)^2$ and that of $(Id - \mathcal{T} \circ \mathcal{S}(.,0))^i$ (for $i \geqslant [N/2]$) by using Theorem 3.

Remark 8. *From formulas* (12), *it appears that under the assumptions of Theorem* 3, *the symbol* $\widehat{\mathcal{S}}(\xi)$ *of the unpreconditioned system* $\mathcal{S}(.,0)$ *has basically the following structure*

$$\widehat{\mathcal{S}}(\xi) = \frac{v}{2}(\lambda^- - \lambda^+)(\xi)\ \left(Id + \widehat{\mathcal{A}}'(\xi) + \widehat{\mathcal{B}}'(\xi)\right),$$

where $\widehat{\mathcal{A}}'$ *is a nilpotent matrix of order* $N-1$ *and* $\widehat{\mathcal{B}}'$ *is small. The factor* $v/2(\lambda^+ - \lambda^-)(\xi)$ *which corresponds in the physical space to an unbounded operator may lead to an ill-conditioned linear system and then to a poor convergence of the GMRES algorithm, see Section* 4.

2.3. The discrete case

We suppose for simplicity that the computational domain is $\mathbb{R}^2$ discretized by a Cartesian grid. Let us denote $A = (A^{kl}_{ij})_{i,j,k,l\in\mathbb{Z}}$ the matrix resulting from a discretization of the advection–diffusion problem. We suppose that the stencil is a 9-point stencil $\left(A^{kl}_{ij} = 0 \text{ for } |i-k| \geqslant 2 \text{ or } |j-l| \geqslant 2\right)$. This is the case, for instance, for a Q1-SUPG method or for a classical finite difference or finite volume scheme. We have to solve $AU = F$, where $U = (u_{ij})_{i,j\in\mathbb{Z}}$ is the vector of the unknowns. The computational domain is decomposed into two half planes ω_1 and ω_2. We introduce a discretized form $\mathcal{S}_h$ of the operator $\mathcal{S}$ (we adopt the summation convention of Einstein over all repeated indices)

$$\begin{aligned}&\mathcal{S}_h : \mathbb{R}^{\mathbb{Z}} \times \mathbb{R}^{\mathbb{Z}\times\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}\\ &((u_{0j})_{j\in\mathbb{Z}}, (F_{ij})_{i,j\in\mathbb{Z}}) \mapsto (A_{0j}^{-1l} v_{-1l}^1 + B_j^{1l} v_{0l}^1 + A_{0j}^{1l} v_{1l}^2 + B_j^{2l} v_{0l}^2 - F_{0j})_{j\in\mathbb{Z}},\end{aligned}$$

where (v_{ij}^m) satisfy

$$\begin{aligned}A_{ij}^{kl} v_{kl}^m &= F_{ij} \quad \text{for } i < 0 \text{ if } m = 1 \quad \text{and} \quad i > 0 \text{ if } m = 2,\ j \in \mathbb{Z},\\ v_{0j}^m &= u_{0j} \quad \text{for } j \in \mathbb{Z}.\end{aligned}$$

The coefficients B_j^{ml} are the contributions of the domain ω_m to A_{0j}^{0l}, $A_{0j}^{0l} = B_j^{1l} + B_j^{2l}$. For example, if $\vec{a} = 0, B_j^{1l} = B_j^{2l} = A_{0j}^{0l}/2$. For example, for a 1D case with a uniform grid and an upwind finite difference scheme $(a > 0)$ and $c = 0$,

$$A_0^{-1} = -\frac{v}{h^2} - \frac{a}{h}, \quad A_0^0 = \frac{2v}{h^2} + \frac{a}{h}, \quad A_0^1 = -\frac{v}{h^2}$$

and

$$B^1 = \frac{v}{h^2} + \frac{a}{h} \quad \text{and} \quad B^2 = \frac{v}{h^2}.$$

The vector $\mathcal{S}_h(u_0, F)$ is the residual of the equation on the interface. It is clear that $U_0 = (u_{0j})_{j\in\mathbb{Z}}$ satisfies

$$\mathcal{S}_h(U_0, 0) = -\mathcal{S}_h(0, F). \tag{20}$$

We propose for an approximate inverse of $\mathcal{S}_h(.,0), \mathcal{T}_h$ defined by

$$\begin{aligned}&\mathcal{T}_h :\ \mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}\\ &(g_j)_{j\in\mathbb{Z}} \mapsto \frac{1}{2}(v_{0j}^1 + v_{0j}^2)_{j\in\mathbb{Z}},\end{aligned}$$

where $(v_{ij}^1)_{i\leqslant 0, j\in\mathbb{Z}}$ and $(v_{ij}^2)_{i\geqslant 0, j\in\mathbb{Z}}$ satisfy

$$A_{ij}^{kl} v_{kl}^1 = 0, \quad i < 0, j \in \mathbb{Z}, \qquad A_{0j}^{-1l} v_{-1l}^1 + \frac{A_{0j}^{0l}}{2} v_{0l}^1 = g_j \tag{21}$$

and

$$A_{ij}^{kl} v_{kl}^2 = 0, \quad i > 0, j \in \mathbb{Z}, \qquad A_{0j}^{1l} v_{1l}^2 + \frac{A_{0j}^{0l}}{2} v_{0l}^2 = g_j. \tag{22}$$

Remark 9. *For a Neumann–Neumann preconditioner, in place of* (21) *and* (22), *we would have* $A_{0j}^{-1l} v_{-1l}^1 + B_j^{1l} v_{0l}^1 = g_j$ *and* $A_{0j}^{1l} v_{1l}^2 + B_j^{2l} v_{0l}^2 = g_j$.

Remark 10. *For a constant coefficient operator* $\mathcal{L}$ *and a uniform grid, a discrete Fourier analysis can be performed similarly to that of Section* 2. *It can then be proved that*

$$\mathcal{T}_h \circ \mathcal{S}_h(.,0) = Id_h.$$

Remark 11. *The last equations of* (21) *and* (22) *correspond to a discretization of the Robin boundary condition* $v(\partial/\partial n) - \vec{a} \cdot \vec{n}/2$. *Considering the previous* 1*D example, we have*

$$h\left(A_0^{-1} v_{-1}^1 + \frac{A_0^0}{2} v_0^1\right) = (v + ah/2)\frac{v_0^1 - v_{-1}^1}{h} - \frac{a}{2} v_{-1}^1 = v\frac{v_0^1 - v_{-1}^1}{h} - \frac{a}{2}(2v_{-1}^1 - v_0^1)$$

and

$$h\left(A_j^{11}v_1^2+\frac{A_0^0}{2}v_0^2\right)=\nu\frac{v_0^2-v_1^2}{h}+\frac{a}{2}v_0^2. \tag{23}$$

Another discretization of $\nu(\partial/\partial n)-\vec{a}\cdot\vec{n}/2$ *would not give* (23). *This is the reason why the approximate inverse is directly defined at the algebraic level. The discretization of the Robin boundary condition is in some sense adaptive with respect to the discretization of the operator* (*SUPG*, *upwind finite difference scheme or finite volume scheme*). *In the previous* 1*D example*, *a straight forward discretization of the Robin boundary condition would give for the first domain*

$$\nu\frac{v_0^1-v_{-1}^1}{h}-\frac{a}{2}v_0^1.$$

When $ah\gg\nu$, *which is usually the case*, *it is quite different from* (21).

In a finite element framework, the variational generalization of the next section gives *automatically* the good discretization of the Robin–Robin boundary condition.

3. Variational generalization

3.1. The continuous problem

Let us now consider general partitions of our original domain Ω into non-overlapping subdomains $\Omega_i, i=1,\dots,N$ (Fig. 1), with interfaces

$$\Gamma_i=\partial\Omega_i\setminus\partial\Omega,\quad \Gamma=\cup_i\Gamma_i.$$

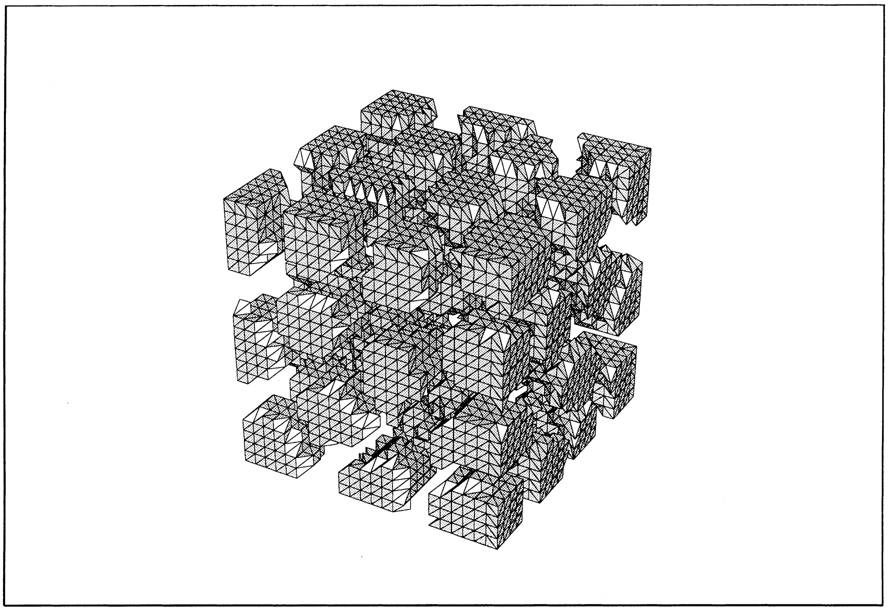

Fig. 1. Three-dimensional triangulation and automatic decomposition into 45 subdomains.

On Ω, we wish to solve the general advection–diffusion problem

$$-\operatorname{div}(\nu(x)\nabla u) + \vec{a}(x)\cdot\nabla u + c(x)u = f \quad \text{on } \Omega, \tag{24}$$

$$u = 0 \quad \text{on } \partial\Omega_D, \tag{25}$$

$$\nu(x)\frac{\partial u}{\partial n} = g \quad \text{on} \quad \partial\Omega_N. \tag{26}$$

We will restrict ourselves to well-posed problems, where the velocity field $\vec{a} \in W^{1,\infty}(\Omega)$ is of bounded divergence

$$c - \frac{1}{2}\operatorname{div}(\vec{a}) \geqslant \alpha > 0$$

for some $\alpha \in \mathbb{R}$ and where the Neumann boundary conditions are only applied on a subset $\partial\Omega_N$ of the domain boundary $\partial\Omega$ where we have outflow conditions (with $\vec{n}$ denoting as usual the outwards pointing unit normal vector to $\partial\Omega$)

$$\vec{a}\cdot\vec{n} \geqslant 0 \quad \forall x \in \partial\Omega_N.$$

The variational framework is better suited for extending the previous sub-structuring technique to such a general formulation and partition. The key point is then to identify the correct bilinear form describing the action of the above non-symmetric advection–diffusion operator on any given subdomain Ω_i of Ω. A naive choice would be to use

$$\hat{a}_i(u,v) = \int_{\Omega_i} \nu\nabla u\cdot\nabla v + (\vec{a}\cdot\nabla u)v + cuv.$$

But in fact, this choice is rather inappropriate, because of its lack of positiveness. A better choice is to integrate by parts the advection term $1/2(\vec{a}(x)\cdot\nabla u)v$ and to introduce the local symmetrized form

$$\begin{aligned} a_i(u,v) &= \int_{\Omega_i}\left\{\nu\nabla u\cdot\nabla v + \frac{1}{2}(\vec{a}\cdot\nabla u)v - \frac{1}{2}(\vec{a}\cdot\nabla v)u + (c - \frac{1}{2}\operatorname{div}(\vec{a}))uv\right\} + \int_{\partial\Omega\cap\partial\Omega_i}\frac{1}{2}\vec{a}\cdot\vec{n}uv \\ &= \hat{a}_i(u,v) - \int_{\Gamma_i}\frac{1}{2}\vec{a}\cdot\vec{n}_i uv. \end{aligned} \tag{27}$$

With this choice, a direct multiplication and integration of (24) yields the variational problem

$$\text{Find } u \in \mathbb{H}(\Omega): \quad \sum_{i=1}^{N}\{a_i(u,v) - L_i(v)\} = 0 \quad \forall v \in \mathbb{H}(\Omega), \tag{28}$$

with the notation

$$\mathbb{H}(\Omega) = \{v \in H^1(\Omega), v = 0 \quad \text{on } \partial\Omega_D\},$$

$$L_i(v) = \int_{\Omega_i} fv + \int_{\partial\Omega_N\cap\partial\Omega_i} gv.$$

In this formulation, the interface terms $-\int_{\Gamma_i} 1/2\vec{a}\cdot\vec{n}_i uv$ added locally to each local form $\hat{a}_i$ cancel each other by summation. Their presence nevertheless guarantee that the local bilinear form a_i is positive on the space of restrictions

$$\mathbb{H}(\Omega_i) = \{v_i = v|_{\Omega_i}, \quad v \in \mathbb{H}(\Omega)\}$$

since we have by construction

$$a_i(u,u) = \int_{\Omega_i}\nu|\nabla u|^2 + \left(c - \frac{1}{2}\operatorname{div}(\vec{a})\right)u^2 + \int_{\partial\Omega_N\cap\partial\Omega_i}\frac{1}{2}\vec{a}\cdot\vec{n}u^2 \quad \forall u \in \mathbb{H}(\Omega_i).$$

Moreover, one can easily observe that the natural interface boundary condition which appears when we integrate this bilinear form in (27) by parts is precisely the Robin condition

$$v(x)\frac{\partial u}{\partial n} - \frac{1}{2}\vec{a}\cdot\vec{n}_i u = \ldots \text{ on } \Gamma_i.$$

used in Section 2.

3.2. Finite element discretization

In view of its numerical solution, the above variational formulation (28) must first be approximated by finite element methods, that is by replacing the space $\mathbb{H}(\Omega)$ by a finite element space $\mathbb{H}_h(\Omega)$. For example, this space is constructed with second order isoparametric finite elements defined on regular "triangulations" of Ω (see Fig. 1).

They are a good compromise between accuracy and cost-efficiency. Other choices of finite element are of course possible, and some of the following numerical tests will use simpler triangular first order elements. In any case, the triangulations to be introduced will respect the subdomain partitions used in domain decomposition: the interfaces Γ_i coincide with inter-elements boundaries, which mean that each subdomain is the union of a given subset of elements of the original triangulation.

When used on advection dominated problems, these finite element techniques must be stabilized. In Galerkin least-squares techniques, this is done by adding element residuals

$$\int_{T_l} \tau_e(h_l)\Big(- \operatorname{div}(v\nabla u) + \vec{a}\cdot\nabla u + cu - f\Big)\Big(- \operatorname{div}(v\nabla v) + \vec{a}\cdot\nabla v + cv\Big)$$

to the original variational formulation with an adequate choice of the local positive stabilization parameters $\tau_e(h_l)$. The stabilized finite element formulation is then

$$\text{Find } u_h \in \mathbb{H}_h(\Omega): \quad \sum_{i=1}^{N}\{a_{ih}(u_h, v_h) - L_{ih}(v_h)\} = 0 \quad \forall v_h \in \mathbb{H}_h(\Omega) \tag{29}$$

with the notation

$$a_{ih}(u,v) = a_i(u,v) + \sum_{T_l\subset\Omega_i}\int_{T_l}\tau_e(h_l)(-\operatorname{div}(v\nabla u) + \vec{a}\cdot\nabla u + cu)\Big(- \operatorname{div}(v\nabla v) + \vec{a}\cdot\nabla v + cv\Big), L_{ih}(v)$$
$$= L_i(v) + \sum_{T_l\subset\Omega_i}\int_{T_l}\tau_e(h_l)f(-\operatorname{div}(v\nabla v) + \vec{a}\cdot\nabla v + cv).$$

We observe that the variational structures of the original problem and of its finite element discretization are very similar. This will also be true for the numerical domain decomposition technique to be introduced next. Therefore, from now on, we will use the same notation for both the continuous and the stabilized discrete problem and omit the subscript h in all finite element formulations. The reader will just have to remember that when dealing with finite elements, the bilinear and linear forms a_i and L_i, and the space $\mathbb{H}(\Omega)$ should be replaced by their discrete counterparts a_{ih}, L_{ih} and $\mathbb{H}_h(\Omega)$ as defined in the present section.

3.3. Substructuring

Due to their variational structure, problems (28) or (29) can be reduced to interface problems by standard substructuring techniques. For this purpose, we introduce the local spaces of restrictions

$$\mathbb{H}(\Omega_i) = \{v_i = v\,|_{\Omega_i},\ v\in\mathbb{H}(\Omega)\},$$
$$\overset{\circ}{\mathbb{H}}(\Omega_i) = \{v_i \in \mathbb{H}(\Omega), v_i = 0 \text{ in } \overline{\Omega\setminus\Omega_i}\},$$
$$= \text{space of functions of } \mathbb{H}(\Omega_i) \text{ with zero continuous extension in } \overline{\Omega\setminus\Omega_i},$$

the global trace space $\mathbb{V} = \mathrm{Tr}\mathbb{H}(\Omega)\,|_\Gamma$, the local trace spaces

$$\mathbb{V}_i = \{\bar{v}_i = \mathrm{Tr}v_i\,|_{\Gamma_i},\quad v_i\in\mathbb{H}(\Omega_i)\} = \{\bar{v}_i = \mathrm{Tr}v\,|_{\Gamma_i},\quad v\in\mathbb{H}(\Omega)\},$$

the a_i-harmonic extension $\mathrm{Tr}_i^{-1} : \mathbb{V}_i \to \mathbb{H}(\Omega_i)$ defined by

$$a_i(\mathrm{Tr}_i^{-1}\bar{u}_i, v_i) = 0 \quad \forall v_i \in \overset{\circ}{\mathbb{H}}(\Omega_i), \qquad \mathrm{Tr}(\mathrm{Tr}_i^{-1}\bar{u}_i)\,|_{\Gamma_i} = \bar{u}_i, \qquad \mathrm{Tr}_i^{-1}\bar{u}_i \in \mathbb{H}(\Omega_i) \tag{30}$$

and its adjoint Tr_i^{-*} given by

$$a_i(v_i, \mathrm{Tr}_i^{-*}\bar{u}_i) = 0 \quad \forall v_i \in \overset{\circ}{\mathbb{H}}(\Omega_i), \mathrm{Tr}(\mathrm{Tr}_i^{-*}\bar{u}_i)\,|_{\Gamma_i} = \bar{u}_i, \mathrm{Tr}_i^{-*}\bar{u}_i \in \mathbb{H}(\Omega_i). \tag{31}$$

By construction, the bilinear form a_i is elliptic on $\overset{\circ}{\mathbb{H}}(\Omega_i)$. Hence, the problems (30) or (31) are well-posed. We then can define the local Schur complement operator $S_i : \mathbb{V}_i \to \mathbb{V}_i'$ by

$$\langle S_i\bar{u}_i, \bar{v}_i\rangle = a_i(\mathrm{Tr}_i^{-1}\bar{u}_i, \mathrm{Tr}_i^{-*}\bar{v}_i) \quad \forall \bar{u}_i, \bar{v}_i \in \mathbb{V}_i.$$

In matrix form, by decomposing the local degrees of freedom U_i of $u_i = R_i u$ (with R_i (resp. $\bar{R}_i$) denoting the restriction operator from $\mathbb{H}(\Omega)$ into $\mathbb{H}(\Omega_i)$ (resp. from $\mathbb{V}$ into $\mathbb{V}_i$)) into internal degrees of freedom $\overset{\circ}{\mathrm{U}}_i$ and interface degrees of freedom $\bar{U}_i$, and the matrix A_i associated to the bilinear form a_i into

$$A_i = \begin{bmatrix} \overset{\circ}{\mathrm{A}}_i & B_i \\ \tilde{B}_i^{\mathrm{T}} & \bar{A}_i \end{bmatrix}, \tag{32}$$

we simply have

$$\mathrm{Tr}_i^{-1} = \begin{pmatrix} -\overset{\circ}{\mathrm{A}}_i^{-1} & B_i \\ I & \end{pmatrix},$$

$$S_i\bar{U}_i = (\bar{A}_i - \tilde{B}_i^{\mathrm{T}}\overset{\circ}{\mathrm{A}}_i^{-1}B_i)\bar{U}_i.$$

With this notation, we can decompose each restriction of the solution u and test function v into $R_i u = \overset{\circ}{\mathrm{u}}_i + \mathrm{Tr}_i^{-1}(\bar{R}_i u)$ and $\bar{R}_i v = \overset{\circ}{\mathrm{v}}_i + \mathrm{Tr}_i^{-*}(\bar{R}_i v)$, then eliminate the local component $\overset{\circ}{\mathrm{u}}_i$ because it is solution of the local well posed problem

$$a_i(\overset{\circ}{\mathrm{u}}_i, v_i) = L_i(v_i) \quad \forall v_i \in \overset{\circ}{\mathbb{H}}(\Omega_i)\ \overset{\circ}{\mathrm{u}}_i \in \overset{\circ}{\mathbb{H}}(\Omega_i). \tag{33}$$

Therefore, we can reduce the original problems (28) or (29) to the interface problem

$$S\bar{u} = F \quad \text{in } \mathbb{V}. \tag{34}$$

Here, we have introduced the global Schur complement operator

$$S = \sum_{i=1}^{N} \bar{R}_i^{\mathrm{T}} S_i \bar{R}_i,$$

and the interface right-hand side F defined by

$$\begin{aligned}\langle F, \bar{v}\rangle &= \sum_i L_i(\mathrm{Tr}_i^{-*}\bar{R}_i\bar{v}) \\ &= \sum_i \{L_i(v_i) - L_i(v_i - \mathrm{Tr}_i^{-*}\bar{R}_i\bar{v})\} \\ &= \sum_i \left\{L_i(v_i) - a_i\left(\overset{\circ}{\mathrm{u}}_i, v_i - \mathrm{Tr}_i^{-*}\bar{R}_i\bar{v}\right)\right\} \left(\text{construction of}\overset{\circ}{\mathrm{u}}_i\right) \\ &= \sum_i \left\{L_i(v_i) - a_i\left(\overset{\circ}{\mathrm{u}}_i, v_i\right)\right\} \left(\text{construction of } \mathrm{Tr}_i^{-*}\right),\end{aligned}$$

with v_i denoting any function of $\mathbb{H}(\Omega_i)$ such that $v_i = \bar{v}$ on Γ_i.

Remark 3.1. *By construction of* Tr_i^{-1}, *the function* $\mathrm{Tr}_i^{-*}\bar{v}_i$ *used in the definition of* S_i *can be replaced by any function* v_i *whose trace on* Γ_i *is equal to* $\bar{v}_i$, *yielding*

$$\langle S\bar{u}_i, \bar{v}\rangle = \sum_i a_i(\mathrm{Tr}_i^{-1}\bar{R}_i\bar{u}, v_i) \quad \forall \bar{u},\ \ \bar{v} \in \mathbb{V}.$$

3.4. Basic preconditioner

The specific preconditioner detailed in this paper is a natural generalization of the so-called Neumann–Neumann algorithm proposed by Agoskhov [2], Bourgat et al. [3], Morice [16], among others. The idea is to precondition the interface operator $S = \sum_i R_i^{\mathrm{T}} S_i R_i$ by a weighted sum of inverses [22]:

$$M^{-1} = T = \sum_i D_i (S_i)^{-1} D_i^{\mathrm{T}}. \tag{35}$$

Above, D_i are diagonal weighting matrices satisfying

$$\sum_i D_i R_i = Id.$$

For implementation reasons (flexibility and parallelism), the map D_i must be as local as possible. The generic choice consists in defining D_i on each interface degree of freedom $\bar{v}(P_l)$ by:

$$D_i \bar{v}(P_l) = \frac{\rho_i}{\rho} \bar{v}(P_k)$$

if the l degree of freedom of $\mathbb{V}$ corresponds to the k degree of freedom of $\mathbb{V}_i$,

$$D_i \bar{v}(P_l) = 0,$$

if not. Here ρ_i is a local measure of the stiffness of subdomain Ω_i (for example the local diagonal value of A_i) and $\rho = \sum_{P_l \in \Omega_j} \rho_j$ is the sum of ρ_j on all subdomains Ω_j containing P_l.

Remark 3.2. *For any* $F_i \in \mathbb{V}_i'$, $(S_i)^{-1} F_i$ *is equal to the trace on* Γ_i *of the solution* w_i *of the local variational problem*

$$a_i(w_i, v_i) = \langle F_i, \mathrm{Tr}(v_i)_{|\Gamma_i}\rangle \quad \forall v_i \in \mathbb{H}(\Omega_i),\ \ w_i \in \mathbb{H}(\Omega_i). \tag{36}$$

By construction of our bilinear form a_i, *this local problem is associated to the operator*

$$-\mathrm{div}(\nu \nabla w) + \vec{a} \cdot \nabla w + cw$$

and to the Robin boundary condition on the interface

$$\nu \frac{\partial w}{\partial n} - \frac{1}{2} \vec{a} \cdot \vec{n}_i w = F_i \quad \text{on } \Gamma_i.$$

By construction we also have (*omitting the positive stabilization term*

$$\sum_{T_l \subset \Omega_i} \int_{T_l} \tau_e(h_l)(-\mathrm{div}(\nu \nabla w) + \vec{a} \cdot \nabla w + cw)^2$$

at the finite element level),

$$a_i(w, w) = \int_{\Omega_i} \nu |\nabla w|^2 + \left(c - \frac{1}{2}\mathrm{div}(\vec{a})\right) w^2 + \int_{\partial\Omega_N \cap \partial\Omega_i} \frac{1}{2} \vec{a} \cdot \vec{n} w^2,$$

which means that the bilinear form a_i *is strictly elliptic on* $\mathbb{H}(\Omega_i)$. *Therefore*, *the variational problem* (36) *is always well posed.*

3.5. Coarse solver

When used on elliptic symmetric operators, it is observed that the basic preconditioner introduced in Section 3.4 behaves poorly when the number of subdomains is large. This is due to the existence of constant like functions which scale badly in energy norm when expanded from a local subdomain to the full domain. The traditional remedy is to build a small coarse global space which includes all these constant like functions and to solve the interface problem by a direct solver on this coarse space, and by a Neumann–Neumann preconditioned iterative solver on its orthogonal. This strategy can also be used for advection problems, although the optimal coarse space is not quite known yet in such problems.

For this purpose, we introduce a given coarse space $\mathbb{V}_G$. Its construction will be done later. Its S orthogonal space $\mathbb{V}_f$ is then defined by

$$\mathbb{V}_f = \{v_f \in \mathbb{V}, \langle Sv_f, \hat{v}_G\rangle = 0 \quad \forall \hat{v}_G \in \mathbb{V}_G\} \tag{37}$$

and the action of the projector $I - P_G$ form $\mathbb{V}$ into $\mathbb{V}_f$ parallel to $\mathbb{V}_G$ on any given function $\bar{v} \in \mathbb{V}$ amounts to the solution of the global coarse variational problem

$$\begin{aligned} &\langle Sv_G, \hat{v}_G\rangle = \langle S\bar{v}, \hat{v}_G\rangle \quad \forall \hat{v}_G \in \mathbb{V}_G, \;\; v_G \in \mathbb{V}_G, \\ &(I - P_G)\bar{v} = \bar{v} - v_G. \end{aligned} \tag{38}$$

Again, the problem (38) is well-posed, since by construction the operator S is elliptic on $\mathbb{V}_G$. Indeed, computing S as proposed in Remark 3.1 with $v_i = \mathrm{Tr}_i^{-1}\bar{R}_i v_G$, and omitting for simplicity the positive stabilization terms, we have

$$\langle Sv_G, v_G\rangle = \sum_i \int_{\Omega_i} v|\nabla \mathrm{Tr}_i^{-1}\bar{R}_i v_G|^2 + \left(c - \frac{1}{2}\mathrm{div}(\vec{a})\right)(\mathrm{Tr}_i^{-1}\bar{R}_i v_G)^2 + \int_{\partial\Omega_N \cap \partial\Omega_i} \frac{1}{2}\vec{a}\cdot\vec{n}\,(\mathrm{Tr}_i^{-1}\bar{R}_i v_G)^2.$$

Because of the presence of a coarse space, the solution $\bar{u}$ of the interface problem can now be uniquely decomposed in

$$\bar{u} = u_G + u_f, \quad u_G \in \mathbb{V}_G, \quad u_f \in \mathbb{V}_f,$$

the coarse component u_G being solution of the global coarse problem

$$\langle Su_G, \hat{v}_G\rangle = \langle F, \hat{v}_G\rangle \quad \forall \hat{v}_G \in \mathbb{V}_G, \;\; u_G \in \mathbb{V}_G, \tag{39}$$

to be solved by a direct solver, and the orthogonal component u_f being the solution of the remaining problem

$$\langle Su_f, \hat{v}_f\rangle = \langle F, \hat{v}_f\rangle - \langle Su_G, \hat{v}_f\rangle \quad \forall \hat{v}_f \in \mathbb{V}_f, \;\; u_f \in \mathbb{V}_f, \tag{40}$$

to be obtained by a GMRES type iterative algorithm with updated preconditioner

$$M^{-1} = (I - P_G)\cdot T \cdot (I - P_G)^{\mathrm{T}} = (I - P_G)\cdot(\Sigma_i D_i (S_i)^{-1} D_i^{\mathrm{T}})\cdot(I - P_G)^{\mathrm{T}}. \tag{41}$$

We can observe that the basic preconditioner is now only used to obtain the component of $\bar{u}$ orthogonal to the coarse space, and therefore avoids as much as possible the bad elements of the coarse space.

Remark 3.3. *The action of $(I - P_G)^{\mathrm{T}}$ is often very easy to compute. Indeed, for any v_f in $\mathbb{V}_f$, we have*

$$\langle (I - P_G)^{\mathrm{T}} Sv_f, \hat{v}\rangle = \langle Sv_f, (I - P_G)\hat{v}\rangle = \langle Sv_f, \hat{v} - \hat{v}_G\rangle = \langle Sv_f, \hat{v}\rangle \quad \forall \hat{v} \in \mathbb{V}.$$

Similarly, by construction of u_G, we also have

$$(I - P_G)^{\mathrm{T}}(F - Su_G) = F - Su_G.$$

3.6. Detailed algorithm

3.6.1. Preparation step

The algorithm involves at each step the solution of Dirichlet problems (30) and Robin problems (36). For each subdomain, these local problems will be solved by a direct method. As the corresponding matrices do not change from one iteration to another, it is possible to compute and factorize them once for ever. Then the solution of the local linear systems will be a simple forward–backward substitution. Similarly, the matrix of the projector (38) can be computed and factorized once for all.

The preparation step therefore executes the following tasks:

1. For each subdomain compute and factorize the matrices A_i (respectively $\mathring{\mathrm{A}}_i$) of the Robin problems (36) (respectively the Dirichlet problems (30)).
2. Compute the interface weighting coefficients D_i.
3. For the coarse problem, for each subdomain (the loop on subdomains has to be replaced by a loop on cross-points when one uses the BPS piecewise linear coarse space):
   - Compute the local functions z^i_α to be used to construct the basis $(D_j\bar{z}^j_\alpha)j, \alpha$ of the coarse space $\mathbb{V}_G$.
   - Compute the vectors $S^{\mathrm{T}}D_j\bar{z}^j_\alpha$. By construction, this means computing $\mathrm{Tr}_i^{-*}D_j\bar{z}^j_\alpha$ by solving the local Dirichlet problems

     $$a_i^{\mathrm{T}}(\mathrm{Tr}_i^{-*}D_j\bar{z}^j_\alpha, v_i) = 0 \quad \forall v_i \ \in \ \mathring{\mathbb{H}}(\Omega_i),$$
     $$\mathrm{Tr}(\mathrm{Tr}_i^{-*}D_j\bar{z}^j_\alpha)_{|\Gamma_i} = D_j\bar{z}^j_\alpha, \ \ \mathrm{Tr}_i^{-*}D_j\bar{z}^j_\alpha \in \mathbb{H}(\Omega_i),$$

     with matrix $(\mathring{\mathrm{A}}_i)^{\mathrm{T}}$ (Using there the available factorization of $\mathring{\mathrm{A}}_i$) and multiple right-hand sides $D_j\bar{z}^j_\alpha$, and setting

     $$\langle S^{\mathrm{T}}D_j\bar{z}^j_\alpha, \bar{v}\rangle = \sum_i a_i^t(\mathrm{Tr}_i^{-*}D_j\bar{z}^j_\alpha, v_i)$$

     where v_i is any consistent extension of $\bar{v}$. If *NEIGH* is the number of neighboring subdomains of domain i, if N_α is the number of local coarse functions, the number of right-hand sides to solve for each subdomain i is equal to $N_\alpha\times$ (*NEIGH* + 1).
   - Compute and factor the coarse matrix $C_{j\beta,i\alpha} = \langle D_i\bar{z}^i_\alpha, S^{\mathrm{T}}D_j\bar{z}^j_\beta\rangle(N_\alpha\times$ (*NEIGH* + 1) dot products).
4. Compute the local components $\mathring{\mathrm{u}}_i$ of the solution by solving the Dirichlet problem(with matrix $\mathring{\mathrm{A}}_i$)

   $$a_i(\mathring{\mathrm{u}}_i, v_i) = L_i(v_i) \quad \forall v_i \ \in \ \mathring{\mathbb{H}}(\Omega_i) = \mathring{\mathrm{u}}_i \in \mathring{\mathbb{H}}(\Omega_i).$$

5. Compute the coarse component u_G of the solution by solving the coarse problem (with matrix C)

   $$\langle u_G, S^{\mathrm{T}}D_j\bar{z}^j_\beta\rangle = \sum_i \{L_i(v_i) - a_i(\mathring{\mathrm{u}}_i, v_i)\} \quad \forall j \quad \forall \beta,$$

   where v_i is any consistent extension of $D_j\bar{z}^j_\beta$, that is any function of $\mathbb{V}_i$ such that $v_i = D_j\bar{z}^j_\beta$ on Γ_i.

Most of these computations are obviously independent on each subdomain and thus can be run in parallel.

3.6.2. GMRES initialization step

(i) Compute the local extension $\mathrm{Tr}_i^{-1}(u_G + u_f^0)(u_f^0 \in \mathbb{V}_f$ is the initial value of u_f and is usually taken to be $u_f^0 = 0)$ by solving the Dirichlet problem (30) with matrix $\mathring{\mathrm{A}}_i$, zero right-hand side, and interface data $u_G + u_f^0$. This step is not necessary in the symmetric case because when A_i is symmetric, we have already computed $\mathrm{Tr}_i^{-1}D_j\bar{z}^j_\alpha = \mathrm{Tr}_i^{-*}D_j\bar{z}^j_\alpha$.

(ii) Compute the initial residual

$$\langle Res_0, \bar{v}\rangle = \sum_i \{L_i(v_i) - a_i(\mathring{\mathrm{u}}_i + \mathrm{Tr}_i^{-1}(u_G + u_f^0), v_i)\}.$$

(iii) Precondition this residual by first computing $\bar{\psi}^0 = \sum_i D_i(S_i)^{-1}D_i^{\mathrm{T}}\mathrm{Res}_0$ by solving the local Robin problems (with matrix A_i)

$$a_i(w_i, v_i) = \langle \mathrm{Res}_0, D_i Tr(v_i)_{|\Gamma_i} \rangle \quad \forall v_i \in \mathbb{H}(\Omega_i), \ \ w_i \in \mathbb{H}(\Omega_i),$$

and by setting $\bar{\psi}^0 = \sum_i D_i Tr(w_i)$.
(iv) Compute finally the preconditional residual $(I - P_G)\bar{\psi}^0 = \bar{\psi}^0 - \bar{\psi}^0_G$ by solving the coarse problem (with matrix C)

$$\langle \psi^0_G, S^{\mathrm{T}} D_j \bar{z}^j_\beta \rangle = \langle \bar{\psi}^0, S^{\mathrm{T}} D_j \bar{z}^j_\beta \rangle \quad \forall j \quad \forall \beta.$$

Most of these computations are obviously independent on each subdomain and thus can be run in parallel.

3.6.3. Solution step

The practical implementation of a preconditioned GMRES type iterative loop for computing u_f requires then the calculation of products

$$(I - P_G) \cdot \left(\sum_i D_i (S_i)^{-1} D_i^{\mathrm{T}} \right) \cdot (I - P_G)^{\mathrm{T}} S v_f$$

for a series of function $v_f \in \mathbb{V}_f$. Computing such products amounts to the following steps:
(i) Compute the solution $\mathrm{Tr}_i^{-1} v_f$ of the Dirichlet problem (30) with matrix A_i and interface data v_f.
(ii) Compute $\mathrm{d}F = (I - P_G)^{\mathrm{T}} S v_f = S v_f$ by

$$\langle \mathrm{d}F, \bar{v} \rangle = \sum_i a_i(\mathrm{Tr}_i^{-1} v_f, \hat{v}_i),$$

with $\hat{v}_i$ denoting any consistent extension of $\bar{v}$ inside Ω_i.
(iii) Compute $\bar{\psi} = \sum_i D_i (S_i)^{-1} D_i^{\mathrm{T}} dF$ by solving the local Robin problems (with matrix A_i)

$$a_i(w_i, v_i) = \langle \mathrm{d}F, D_i Tr(v_i)_{|\Gamma_i} \rangle \quad \forall v_i \in \mathbb{H}(\Omega_i), \ \ w_i \in \mathbb{H}(\Omega_i),$$

and by setting $\bar{\psi} = \sum_i D_i Tr(w_i)$.
(iv) Compute finally $(I - P_G)\bar{\psi} = \bar{\psi} - \bar{\psi}_G$ by solving the coarse problem (with matrix C)

$$\langle \psi_G, S^{\mathrm{T}} D_j \bar{z}^j_\beta \rangle = \langle \bar{\psi}, S^{\mathrm{T}} D_j \bar{z}^j_\beta \rangle \quad \forall j \quad \forall \beta.$$

Again, the most memory and time consuming steps in this algorithm are independent on each subdomain and thus can be run in parallel.

3.7. Construction of the coarse space

There are two main strategies for constructing coarse spaces. A multigrid based strategy will take as coarse space the trace of low order finite element functions defined on a much coarser triangulation, namely the triangulation constructed by assimilating every subdomain to a coarse finite element. This BPS technique (named after the celebrated paper where it was first analysed) is rather efficient for 2D problems, and functions of the corresponding coarse space are linear on each edge.

A subdomain based strategy will try to identify the possible local pathologies of the Robin preconditioner. We want in particular to deal with positive definite preconditioned operators, that is we would like to have

$$\langle S v_f, M^{-1} S v_f \rangle \geqslant C \|v_f\|^2 \quad \forall v_f \in \mathbb{V}_f,$$

for a proper choice of norm $\|\cdot\|^2$. But a direct calculation yields

$$\begin{aligned}\langle Sv_f, M^{-1}Sv_f\rangle &= \langle Sv_f, (I-P_G)\cdot T\cdot (I-P_G)^{\mathrm{T}}Sv_f\rangle\\ &= \langle (I-P_G)^{\mathrm{T}}Sv_f, T\cdot (I-P_G)^{\mathrm{T}}Sv_f\rangle\\ &= \langle Sv_f, TSv_f\rangle \text{ (construction of } \mathbb{V}_f)\\ &= \sum_i \langle Sv_f, D_i(S_i)^{-1}D_i^{\mathrm{T}}Sv_f\rangle \text{ (construction of } T)\\ &= \sum_i \langle Sv_f, D_i\mathrm{Tr}w_{i|\Gamma_i}\rangle \text{ (construction of } S_i^{-1})\\ &= \sum_i a_i(w_i,w_i) \text{ (construction of } w_i),\end{aligned}$$

where w_i denotes the solution of the local Robin problem (36) solved in the preconditioning step. By construction of our bilinear form, we also have (omitting the stabilization terms)

$$a_i(w_i,w_i) = \int_{\Omega_i} \nu|\nabla w_i|^2 + \left(c - \frac{1}{2}\mathrm{div}(\vec{a})\right)w_i^2 + \int_{\partial\Omega_N\cap\partial\Omega_i} \frac{1}{2}\vec{a}\cdot\vec{n}w_i^2.$$

The situation is therefore bad on internal subdomains (where $\partial\Omega\cap\partial\Omega_i=\emptyset$), when w_i is a constant, and $\int_{\Omega_i}(c-(1/2)\mathrm{div}(\vec{a}))$ is small. To avoid such bad situations, we must forbid nonzero constant solutions in the local Robin problems, which we achieve if we impose that all solutions of the local Robin problems be of zero average. To do this, we take as coarse space the space

$$\mathbb{V}_G = \left\{ v_G, v_G = \sum_i c^i D_i \mathrm{Tr}(v_i^0)_{|\Gamma_i}, (c^i)_i \text{ arbitrary} \right\} \tag{42}$$

generated by the weighted traces $D_i\mathrm{Tr}(v_i^0)_{|\Gamma_i}$ of the solutions v_i^0 of the local adjoint Dirichlet problems

$$a_i^t(v_i^0,\hat{v}_i) = \int_{\Omega_i} \hat{v}_i \;\forall \hat{v}_i \in \mathbb{H}(\Omega_i),\;\; v_i^0 \in \mathbb{H}(\Omega_i).$$

Then, by construction of the orthogonal space V_f, we have

$$\langle Sv_f, \hat{v}_G\rangle = 0 \quad \forall \hat{v}_G \in \mathbb{V}_G \quad \forall v_f \in \mathbb{V}_f.$$

By taking $\hat{v}_G = D_i\mathrm{Tr}(v_i^0)_{|\Gamma_i}$, this implies in particular that the local solution w_i of the local Robin problems satisfy

$$\begin{aligned}\int_{\Omega_i} w_i &= a_i^t(v_i^0,w_i)(\text{construction of } v_i^0) = a_i(w_i,v_i^0) = \langle Sv_f, D_i\mathrm{Tr}(v_i^0)_{|\Gamma_i}\rangle)(\text{construction of } w_i)\\ &= 0(\text{definition of } \mathbb{V}_f),\end{aligned}$$

which is the desired effect.

Remark 3.4. *In the symmetric case*, ($\vec{a}=0$, *c positive arbitrary*), *the local coarse functions* v_i^0 *are simply the constant* $v_i^0 = 1/c$.

4. Numerical results in two-dimensions

The advection–diffusion is discretized on a Cartesian grid by a Q1-streamline-diffusion method ([14]). The interface problem (20) is solved by a preconditioned GMRES algorithm and the stopping criterion is to reduce the initial residual by a factor 10^{-10}. The preconditioners are either of the type Robin–Robin (R–R), Neumann–Neumann (N–N) or the identity (–).

4.1. A first comparison between the preconditioners

The first series of tests will be to compare the performances of the Robin–Robin (R–R), Neumann–Neumann (N–N) preconditioners, and the non-preconditioned method, in some typical situations. No coarse space solver is used here. In these tests, the domain is the rectangle $[0,1] \times [0,0.2]$ partitioned into five square vertical strips of sizes 0.2×0.2. In each subdomain, there is a uniform grid of 60×60 quadrangular elements. We choose $c = 1$ and $\nu = 0.001$ or $\nu = 1$, and four velocities:

(i) $\vec{a} = \vec{e}_1$. In this case, the velocity is perpendicular to the interfaces between the subdomains.

(ii) $\vec{a} = \vec{e}_2$. In this case, the velocity is parallel to the interfaces.

(iii) $\vec{a} = \sqrt{2}/2(\vec{e}_1 + \vec{e}_2)$. We refer to this convecting field as *oblique*.

(iv) $\vec{a} = 2\pi((x_1 - 0.5)\vec{e}_2 - (x_2 - 0.1)\vec{e}_1)$. Here, the convecting field can be seen as a vortex.

Based on these results, several remarks can be made:

- The Robin–Robin preconditioner performs much better than the Neumann–Neumann preconditioner when the viscosity is small while the performances are equivalent for large viscosity.
- For small viscosities, when the velocity is not parallel to the interface, the Neumann–Neumann preconditioner gives very poor results (poorer than when no preconditioner is used). On the contrary, when the velocity is parallel to the interfaces, both the Neumann–Neumann and the Robin–Robin preconditioners work very well (2 iterations), (note that they are equivalent in this case). These results are in complete agreement with the Fourier analysis above.
- When the velocity is normal or oblique, the Robin–Robin preconditioner has nilpotency properties which will be better illustrated in the next series of tests: this explains why the number of iterations is only 3 and 5 for these cases.
- Thus, the Robin–Robin adapts smoothly to the different situations presented in Table 1.
- In our implementation, one iteration of the Robin–Robin or Neumann–Neumann method costs twice as much as when no preconditioner is used. Even though, the Robin–Robin preconditioner remains always cheaper.

4.2. Influence of the number of subdomains: case of strips

The purpose of this series of tests is to assess the nilpotency properties of the algorithm. To illustrate the theory above, the domain is partitioned into vertical strips, the velocity is uniform and normal to the interfaces. The dependence of the number of iterations with respect to the number of strips will be tested.

Here, the domain is $\Omega = (0, N_{\mathrm{sd}}/4) \times (0,1)$, where N_{sd}. It is partitioned into N_{sd} square subdomains, which are rectangles of sizes 0.25×1. In each subdomain, there is a uniform grid of size 20×40.

The velocity is $\vec{a} = 3\overrightarrow{e_1}, c = 1$, and the viscosity is $\nu = 0.001$.

Here again, the three methods are tested (see Table 2).

In Fig. 2, the convergence of the GMRES algorithms with and without the Robin–Robin preconditioner are plotted. It is very clear on the Fig. 2 that for the Robin–Robin preconditioner, the residual does not change significantly for $N_{\mathrm{sd}}/2 = 18$ iterations, then drops quickly. For GMRES without preconditioner, the same kind of phenomenon is observed, but the decay of the residual is obtained only after $N_{\mathrm{sd}} = 36$ iterations. This confirms very well the theory above on the idempotency of the preconditioned or not Schur

Table 1
Number of iterations for different velocity fields and five subdomains

Viscosity	Precond.\velocity	Normal	Parallel	Oblique	Rotating
$\nu = 0.001$	R–R	3	2	5	36
	N–N	52	2	42	>100
	–	14	34	13	71
$\nu = 1$	R–R	9	9	10	10
	N–N	9	9	10	11
	–	30	38	41	41

Table 2
Influence of the number of subdomains

Partition Grid		4×1 20×40	8×1 20×40	12×1 20×40	24×1 20×40	36×1 20×40
$c = 1$, $\vec{a} = 3\vec{e}_1$, $\nu = 0.001$	R–R	5	8	12	23	30
	–	13	18	22	33	44
	N–N	43	>100	>100	>100	>100

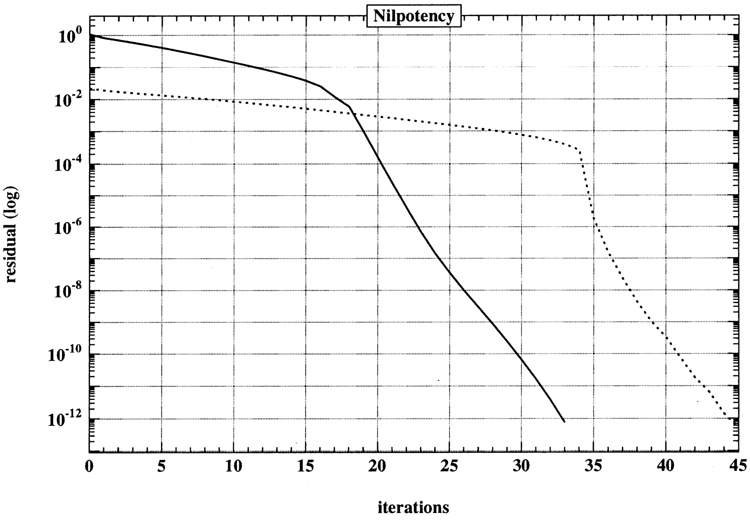

Fig. 2. The convergence with and without preconditioner: 36 subdomains

complement matrix. Note that, since the number of grid nodes is not large enough, the rates of convergence after the idempotency threshold are of the same order. This would not be the case for a finer grid, as we can see on Fig. 3 and in Section 4.3.

On Fig. 3, the convergence plots correspond to the same experiment except that the grid has been refined with a geometrical progression of ratio 0.9 in the x_2 direction. Therefore, the grid is very fine near the boundary $x_2 = 0$.

We see on Fig. 3 that the idempotency properties of the preconditioned operator are conserved, although they appear less clearly. On the contrary, the non-preconditioned operator yields a very poor convergence.

4.3. *Influence of the grid anisotropy and boundary layers*

We carry out exactly the same test as in 4.2, but the velocity has a boundary layer:

$$\begin{aligned} &\vec{a} = 3 - (300 \ * \ (x_2 - 0.1)^2)\vec{e}_1 && \text{if } x_2 < 0.1 \\ &\vec{a} = 3\vec{e}_1 && \text{if } x_2 \geqslant 0.1 \end{aligned} \tag{43}$$

To capture the boundary layer, the mesh is refined in the x_2-direction, near the boundary $x_2 = 0$ with a geometric progression of ratio 0.9 (see Table 3).

In comparison with the tests of Section 4.2, we see that the performances deteriorate due to the change of velocity and grid, but is very clear that the Robin–Robin method is the less severely affected. The nilpotency threshold is also observed here. Table 4 completes these results. Here, the subdomains are squares of sizes 0.25×0.25 and the velocity can either be uniform $\vec{a} = 3\vec{e}_1$ or have a boundary layer, see (43). The meshes can either be uniform, or be progressively refined with a geometrical progression of ratio 0.9 in the x_2 direction. Here again, we see that preconditioning becomes particularly useful when the grid is dramatically refined, as in the case of a boundary layer for instance.

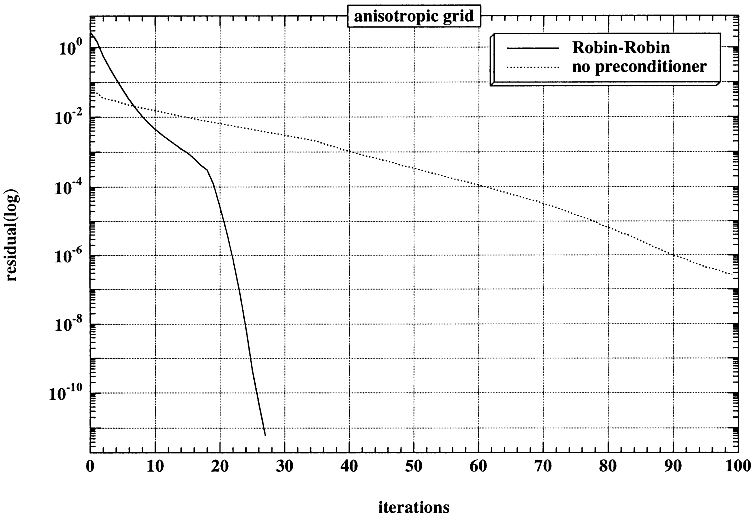

Fig. 3. The convergence with and without preconditioner: 36 subdomains

Table 3
Anisotropic grids

Partition Grid		4 × 1 20 × 40	8 × 1 20 × 40	12 × 1 20 × 40	24 × 1 20 × 40	36 × 1 20 × 40
$c = 1$	R–R	11	18	25	39	51
$\nu = 0.001$	–	51	75	91	> 100	> 100
	N–N	49	> 100	> 100	> 100	> 100

Table 4
Influence of anisotropic mesh refinement and boundary layers

		Partition	3 × 1	3 × 1	3 × 1
Velocity	Refinement		40 × 40	60 × 60	60 × 60
Uniform	Geometrical	R–R	7	7	8
	Progression	–	37	48	69
		R–R	5	5	6
	Uniform	–	12	10	12
B. Layer	Geometrical	R–R	9	8	10
	Progression	–	40	57	73
		R–R	8	7	10
	Uniform	–	18	20	24

4.4. Influence of the number of unknowns

Here, we test the influence of the number of grid points. The domain is the unit square, which is partitioned into 4 × 4 subdomains. In each subdomain, the grid varies from 20 × 20 up to 60 × 60. The velocity is $\vec{a} = \vec{e}_3 \times (\vec{x} - \vec{x}_0)$, ($x_0$ is the center of Ω) the viscosity is $\nu = 0.001$ and the constant c equals to 10^{-7}. Thus, this case is almost a steady state computation. A coarse space solver of BPS type is used here, but it will be discussed later. Note that this test is reputed difficult, since although convection effects dominate, diffusion is the only effect that carries information from the boundary of the domain to its center. The convergence is not affected by the grid size (see Table 5).

Table 5
Influence of the number of grid points

	Partition Grid	4×4 20×20	4×4 30×30	4×4 40×40	4×4 60×60
$c = 10^{-7}$, $v = 0.001$, $\vec{a} = \vec{e}_3 \times (\vec{x} - \vec{x}_0)$	R–R	34	34	34	34
	–	80	82	89	–

Table 6
Coarse Space

Grid	Coarse sp.	2×2 30×30	3×3 30×30	5×5 30×30	7×7 30×30	10×10 30×30
$c = 10^{-4}$	Yes	20	26	36	42	48
$\vec{a} = \vec{e}_3 \times (\vec{x} - \vec{x}_0)$	No	22	31	51	68	91
$c = 10^{-4}$	Yes	7	13	18	22	25
$\vec{a} = 3\vec{e}_1$	No	7	12	15	18	23

4.5. Coarse space

Here, we investigate the possibility of adding a coarse space solver to our preconditioner: we have chosen the space of functions which are the *harmonic* liftings of piecewise linear functions on the skeleton of the domain decomposition. This coarse space is similar to that used by Bramble, Paschiak and Schatz. In addition, for the Robin problems, we fix one internal value to 0 in each subdomain which does not touch the physical boundary. The domain is the unit square, the viscosity is $v = 0.01$ and the parameter c equals to 10^{-4} (the problem is almost stationary). The velocity can either be $\vec{a} = \vec{e}_3 \times (\vec{x} - \vec{x}_0)$ (vortex) or $\vec{a} = 3\vec{e}_1$. The number of subdomains varies from 2×2 up to 10×10, and the grid in each subdomain is 30×30.

Note that the coarse space solver is really useful in the case of the rotating velocity: the convergence still varies when the diameter H of the subdomains decreases but the number of necessary iteration seems to grow sub-linearly with $1/H$, while the growth is linear when no coarse space is used. For the uniform velocity, the coarse space solver does not seem useful. Remark that the viscous effects are more important for the rotating velocity. Further investigations are clearly needed at this level (see Table 6).

5. Preliminary results in three-dimensions

The three-dimensional cases described here deal with domains of arbitrary shapes, decomposed into hexaedra or tetrahedra. The problem (24) is discretized by the stabilized Galerkin Least-Squares technique described in Section 3.2 using second order elements. As in the two-dimensional case, the interface problem is solved by a GMRES algorithm using the Robin–Robin preconditioner. The algorithm stops when the residual is less than 10^{-6}.

5.1. A model problem

The first experiments deal with a regular discretization of the unit cube $[0,1] \times [0,1] \times [0,1]$ on which a problem with a known analytical solution $(u = x^2 + y^2 + z^2)$ is solved. We chose $c = 0.1$ and $v = 1.$ and a uniform oblique velocity $\vec{a} = (1,7,9)$. The total number of finite elements (NE), the corresponding number degrees of freedom (NDOF) and the number of iterations are given in Table 7.

For the decomposition in nine subdomains we have also considered two types of boundary conditions:

- Dirichlet condition on all the boundaries (18 iterations);
- Dirichlet condition on five faces of the cube and Robin boundary condition on the bottom face (40 iterations).

Table 7
A model three-dimensional problem

Partition	2 × 1 × 1	2 × 2 × 1	3 × 3 × 1	3 × 3 × 3
NE	216	256	576	1728
NDOF	1232	1409	3021	8294
R–R	3	10	11	18

No coarse space is used here. The results are sensitive to the number of subdomains, but the number of iterations remains very reasonable in all cases.

5.2. *Influence of the velocity field*

We next study the influence of the velocity field on the behavior of the preconditioner for an advection dominated case. For this we consider the following three decompositions (note that the domain is chosen such as each subdomain has a nice aspect ratio, and that each subdomain is made of 64 finite elements):

Case 1: The cube $[0,1]^2 \times [0,0.5]$ split in $2 \times 2 \times 1$ subdomains

Case 2: The cube $[0,1]^2 \times [0,0.33]$ split in $3 \times 3 \times 1$ subdomains

Case 3: The cube $[0,1]^3$ split in $3 \times 3 \times 3$ subdomains

The results are summarized in Table 8, where $\vec{v}_1$ is defined by:

$$\begin{array}{ll} \vec{v}_1 = 300 \, * \, {x_2}^2 \vec{e}_1 & \text{if } x_2 < 0.1 \\ \vec{v}_1 = 3\vec{e}_1 & \text{if } x_2 \geqslant 0.1 \end{array} \tag{44}$$

and $\vec{v}_2$ denotes the velocity field given by equation (43).

These results show that the Robin–Robin preconditioner is far less sensitive than the original Neumann–Neumann one to the choice of velocity fields. This was expected since the original Neumann–Neumann preconditioner ignores all $\vec{a} \cdot \vec{n}$ boundary coefficients.

Finally we solved the same problem, with $\vec{a} = (y/2 - 0.5, -x/2 + 0.5, 0)$ on the unstructured decomposition of Fig. 1. Here the unit cube contains 24 576 tetrahedric second order finite elements and is split into 45 subdomains by an automatic mesh partitioner. This is why the boundaries between subdomains are less regular than for the other computations. For this decomposition the algorithm converges in 48 iterations with the R–R preconditioner and in 45 iterations with the addition of the coarse space.

Table 8
Influence of the velocity field

	Case 1	Case 2	Case 3
$\vec{a} = (y/2 - 0.5, -x/2 + 0.5, 0), \nu = 10^{-3}, c = 10^{-4}$			
R–R	12	21	35
N–N	24	51	165
$\vec{a} = \vec{v}_1 \, , \; \nu = 10^{-2} \, , \; c = 10^{-4}$			
R–R	9	13	19
N–N	23	65	92
$\vec{a} = \vec{v}_2 \, , \; \nu = 10^{-2} \, , \; c = 10^{-4}$			
R–R	9	16	25
N–N	37	88	>200

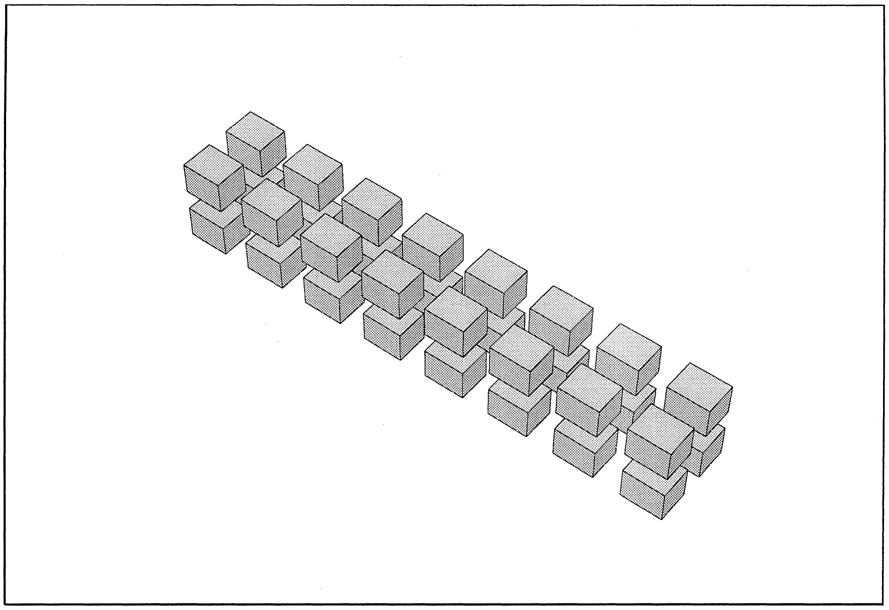

Fig. 4. The decomposition of the tube into subdomains

5.3. *Influence of the coarse space*

In this section we consider a long tube (cf. Fig. 4) $[-2,2]^2 \times [0,2]$ decomposed into 32 subdomains ($2 \times 2 \times 8$).

Each subdomain contains 64 second order hexaedra (425 nodes). The coarse space is the domain based space described in Section 3. We solve the following problem:

$$-\operatorname{div}(v(x)\nabla u) + \vec{a}(x) \cdot \nabla u + c(x)u = 0 \quad \text{on } \Omega, \tag{45}$$

$$u = 2 + x + y \quad \text{on } \partial\Omega_B, \tag{46}$$

$$u = 0 \quad \text{on } \partial\Omega_T, \tag{47}$$

$$v(x)\frac{\partial u}{\partial n} = 0 \quad \text{on } \partial\Omega_N, \tag{48}$$

where Ω_B denotes the bottom of the beam, Ω_T the top and Ω_N the lateral boundary of the beam.

We consider two velocity fields:

$$\vec{a}_1 = 0.1 * y * (x^2 - 4)\vec{e}_1 - 0.1 * x * (y^2 - 4)\vec{e}_2 + 4\vec{e}_3,$$

$$\vec{a}_2 = 0.1 * y * (x^2 - 4)\vec{e}_1 - 0.1 * x * (y^2 - 4)\vec{e}_2 + (4 + 0.1 * z * (y/10 + x/10 - 0.5))\vec{e}_3.$$

The case is difficult because of the aspect ratio of the domain, and because of our choice of coefficients imposing very little diffusion and almost no stabilization c. The results, obtained for different values of v and c are summarized in Table 9.

The coarse space slightly improves the efficiency of the preconditioner, but this is not really spectacular.

The next test is a three-dimensional simulation of the two-dimensional problem solved in the previous section, with the rotating velocities. Starting from a uniform discretization of the unit square in 4 × 4 subdomains we obtain a three-dimensional mesh by transforming each quadrangle into a hexaedron. In order to obtain a good aspect ratio, the third-dimension is equal to the discretization step. Dirichlet boundary conditions are considered on the lateral boundary. The top and bottom faces are subjected to homogeneous Neumann boundary conditions. The results obtained are described in Table 10 for different discretization steps.

Table 9
Influence of the coarse space

Velocity	ν	c	R–R	R–R + coarse
$\vec{a}_1$	10^{-2}	10^{-3}	30	28
$\vec{a}_2$	10^{-2}	10^{-3}	36	31
$\vec{a}_2$	10^{-3}	10^{-7}	57	51

Table 10
Three-dimensional simulation of a two-dimensional problem

Grid	ν	c	R–R	R–R + coarse
4 × 4	10^{-3}	10^{-7}	19	18
20 × 20	10^{-3}	10^{-7}	20	18
20 × 20	10^{-3}	10^{-4}	20	18

The results are very similar to the bidimensional case and completely insensitive to the chosen grid. Moreover, for such a small number of subdomains, the proposed preconditioner is already very efficient without coarse space.

6. Conclusion

The proposed preconditioner is a generalization of the Neumann–Neumann algorithm. Its main advantages are

- Its robustness, assessed first theoretically by a Fourier analysis in the special case of domain decomposition into strips, then by various numerical tests in two and three-dimensions. In the tests above, the Robin–Robin preconditioner has proved fairly insensitive to the choice of the grid, the velocity, the viscosity and the constant c. It is sensitive to the sizes of the subdomains, but this seems unavoidable when convections dominates. Another nice feature is that the Robin–Robin preconditioner transforms continuously to the Neumann–Neumann preconditioner as the convection effects vanish.
- It has the same algebraic structure as the Neumann–Neumann preconditioner. This means that when a software contains the Neumann–Neumann preconditioner, constructing the Robin–Robin preconditioner requires almost no modifications. Therefore, it has been easily implemented in the most general case i.e. a three-dimensional case with unstructured and automatically partitioned mesh.
- A theoretical framework for adding a coarse space solver is now available.

However, our knowledge on this preconditioner is not complete. In particular, the analysis of convergence in the most general case seems difficult, since energy considerations are clearly not sufficient. Further works need to be done, in particular,

- the algorithm will be tested on less academic situations.
- The coarse space to be used will be specified better and validated.

References

[1] Y. Achdou, F. Nataf, A Robin–Robin preconditioner for an advection–diffusion problem, C.R. Acad. Sciences Paris, series I 325 (1997) 1211–1216.

[2] V. Aghoskov, Poincaré–Steklov's operators and domain decomposition methods in finite dimensional spaces, in: Proceedings of DDM 1.

[3] J.F. Bourgat, R. Glowinski, P. Le Tallec, M. Vidrascu, Variational formulation and algorithm for trace operator in domain decomposition calculations, in: Proceedings of DDM 2, AMS, 1988, pp. 3–16.

[4] J. Bramble, J. Pasciak, A. Schatz, The construction of preconditioners for elliptic problems by substructuring, Math. Comp. 47 (1986) 103–134.

[5] Xiao-Chuan Cai, Some Domain Decomposition Algorithms for Nonselfadjoint Elliptic and Parabolic Partial Differential Equations. PhD thesis, Courant Institute of Mathematical Sciences, September 1989, Tech. Rep. 461, Department of Computer Science, Courant Institute.

[6] Xiao-Chuan Cai, An additive Schwarz algorithm for nonselfadjoint elliptic equations, in: T. Chan, R. Glowinski, J. Périaux, O. Widlund (Eds.), Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA, 1990, pp. 232–244.

[7] Xiao-Chuan Cai, O. Widlund, Domain decomposition algorithms for indefinite elliptic problems, SIAM J. Sci. Statist. Comput. 13 (1) (1992) 243–258.

[8] T. Chan, T. Mathew, Domain decomposition algorithms, in: Acta Numerica, Cambridge University Press, Cambridge, 1994, pp. 13–51.

[9] L. Cowsar, J. Mandel, M. Wheeler, Balancing domain decomposition for mixed finite elements, Technical Report 93–08, Department of Mathematical Sciences, Rice University, 1993.

[10] B. Despres, Domain decomposition method and the Helmholtz problem ii, in: K. Ralph et al. (Eds.), Mathematical and Numerical Aspects of Wave Propagation, SIAM, Philadelphia 1993, pp. 197–206.

[11] M. Dryja, O. Widlund, Additive Schwarz methods for elliptic finite element problems in three-dimensions, in: Proceedings of DDM 5.

[12] F. Gastaldi, L. Gastaldi, A. Quarteroni, Adaptative domain decomposition methods for advection dominated equations, East–West J. Numer. Math. 4 (1996) 165–206.

[13] C. Japhet, Optimized Krylov–Ventcell method application to convection–diffusion problems. in: Ninth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Wiley, New York, 1996.

[14] C. Johnson, U. Navert, J. Pitkaranta, Finite element method for linear hyperbolic problems, Comp. Meth. Appl. Engrg. 45 (1984) 285–312.

[15] J. Mandel, Two-level domain decomposition preconditioning for the *p*-version finite element method in three-dimensions, Int. J. Num. Meth. Eng. 29 (1990) 1095–1108.

[16] P. Morice, Transonic computations by a multidomain technique with potential and Euler solvers, in: J. Zienep, H. Oertel (Eds.), Symposium Transsonicum III, IUTAM, Göttingen, 24–27 May 1988, 1989, Springer, Berlin.

[17] F. Nataf, F. Nier, Convergence rate of Schwarz type methods for an arbitrary number of subdomains, in: Ninth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Wiley, New York, 1996.

[18] F. Nataf, F. Nier, Convergence rate of some domain decomposition methods for overlapping and non-overlapping subdomains, Numerische Mathematik 75 (1997) 357–377.

[19] F. Nataf, F. Rogier, Factorization of the convection-diffusion operator and the Schwarz algorithm, Math. Models Appl. Sci. 5 (1995) 67–93.

[20] Y. Saad, H. Schultz, Gmres: a generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Stat. Comput 7 (1986) 856–869.

[21] B. Smith, An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems, SIAM J. Sci Stat. Comp. 13 (1992) 364–378.

[22] P. Le Tallec, Domain decomposition methods in computational mechanics, Comput. Mech. Advances 1 (2) (1994) 121–220.

[23] P. Le Tallec, Y.H. De Roeck, M. Vidrascu, Domain decomposition methods for large linearly elliptic three-dimensional problems, J. Comp. Appl. Math. 34 (1991) 93–117.

[24] O. Widlund, Iterative substructuring methods: algorithms and theory for elliptic problems in the plane, in: G. Golub, R. Glowinski, J. Periaux (Eds.), Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, 1988.