18

Domain decomposition methods

In this chapter we will introduce the domain decomposition method (DD, in short). In its most common version, DD can be used in the framework of any discretization method for partial differential equations (such as, e.g. finite elements, finite volumes, finite differences, or spectral element methods) to make their algebraic solution more efficient on parallel computer platforms. In addition, DD methods allow the reformulation of any given boundary-value problem on a partition of the computational domain into subdomains. As such, it provides a very convenient framework for the solution of *heterogeneous* or multiphysics problems, i.e. those that are governed by differential equations of different kinds in different subregions of the computational domain.

The basic idea behind DD methods consists in subdividing the computational domain Ω, on which a boundary-value problem is set, into two or more subdomains on which discretized problems of smaller dimension are to be solved, with the further potential advantage of using parallel solution algorithms. More in particular, there are two ways of subdividing the computational domain into subdomains: one with disjoint subdomains, the others with overlapping subdomains (for an example, see Fig. 18.1). Correspondingly, different DD algorithms will be set up.
For reference lectures on DD methods we refer to [BGS96, QV99, TW05].

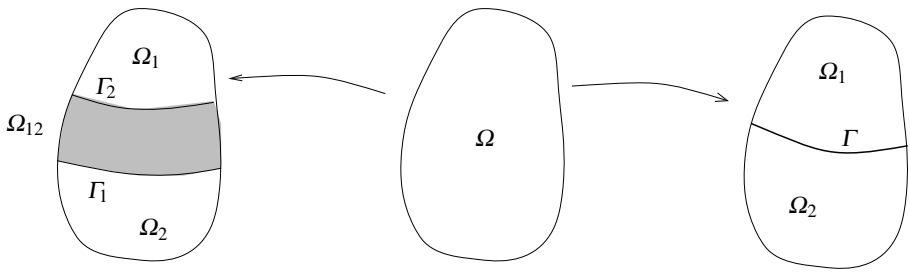

Fig. 18.1. Two examples of subdivision of the domain Ω, with and without overlap

A. Quarteroni: *Numerical Models for Differential Problems*, 2nd Ed.
MS&A – Modeling, Simulation & Applications 8
DOI 10.1007/978-88-470-5522-3_18, © Springer-Verlag Italia 2014

18.1 Some classical iterative DD methods

In this section we introduce four different iterative schemes starting from the model problem: find $u : \Omega \to \mathbb{R}$ such that

$$\begin{cases} Lu = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{18.1}$$

L being a generic second order elliptic operator, whose weak formulation reads

$$\text{find } u \in V = H_0^1(\Omega) \ : \ a(u,v) = (f,v) \quad \forall v \in V, \tag{18.2}$$

being $a(\cdot,\cdot)$ the bilinear form associated with L.

18.1.1 Schwarz method

Consider a decomposition of the domain Ω in two subdomains Ω_1 and Ω_2 such that $\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2$, $\Omega_1 \cap \Omega_2 = \Omega_{12} \neq \emptyset$ (see Fig. 18.1) and let $\Gamma_i = \partial\Omega_i \setminus (\partial\Omega \cap \partial\Omega_i)$. Consider the following iterative method. Given $u_2^{(0)}$ on Γ_1, solve the following problems for $k \geq 1$:

$$\begin{cases} Lu_1^{(k)} = f & \text{in } \Omega_1, \\ u_1^{(k)} = u_2^{(k-1)} & \text{on } \Gamma_1, \\ u_1^{(k)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma_1, \end{cases} \tag{18.3}$$

$$\begin{cases} Lu_2^{(k)} = f & \text{in } \Omega_2, \\ u_2^{(k)} = \begin{cases} u_1^{(k)} \\ u_1^{(k-1)} \end{cases} & \text{on } \Gamma_2, \\ u_2^{(k)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma_2. \end{cases} \tag{18.4}$$

In the case in which one chooses $u_1^{(k)}$ on Γ_2 in (18.4) the method is named *multiplicative Schwarz*, whereas that in which we choose $u_1^{(k-1)}$ is named *additive Schwarz*. The reason will be clarified in Sect.18.6. We have thus two elliptic boundary-value problems with Dirichlet conditions for the two subdomains Ω_1 and Ω_2, and we would like the two sequences $\{u_1^{(k)}\}$ and $\{u_2^{(k)}\}$ to converge to the restrictions of the solution u of problem (18.1), that is

$$\lim_{k\to\infty} u_1^{(k)} = u_{|\Omega_1} \text{ and } \quad \lim_{k\to\infty} u_2^{(k)} = u_{|\Omega_2}.$$

It can be proven that the Schwarz method applied to problem (18.1) always converges, with a rate that increases as the measure $|\Omega_{12}|$ of the overlapping region Ω_{12} increases. Let us show this result on a simple one-dimensional case.

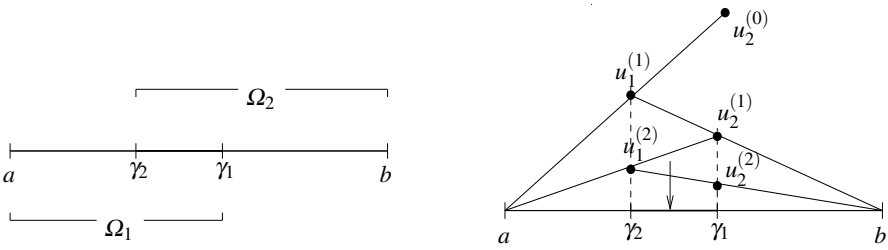

Fig. 18.2. Example of a decomposition with overlap in dimension 1 (left). A few iterations of the multiplicative Schwarz method for problem (18.7) (right)

Example 18.1. Let $\Omega = (a,b)$ and let $\gamma_1, \gamma_2 \in (a,b)$ be such that $a < \gamma_2 < \gamma_1 < b$ (see Fig. 18.2). The two problems (18.3) and (18.4) become:

$$\begin{cases} Lu_1^{(k)} = f, & a < x < \gamma_1, \\ u_1^{(k)} = u_2^{(k-1)}, & x = \gamma_1, \\ u_1^{(k)} = 0, & x = a, \end{cases} \tag{18.5}$$

$$\begin{cases} Lu_2^{(k)} = f, & \gamma_2 < x < b, \\ u_2^{(k)} = u_1^{(k)}, & x = \gamma_2, \\ u_2^{(k)} = 0, & x = b. \end{cases} \tag{18.6}$$

To show that this scheme converges, let us bound ourselves to the simpler problem

$$\begin{cases} -u''(x) = 0, & a < x < b, \\ u(a) = u(b) = 0, \end{cases} \tag{18.7}$$

that is the model problem (18.1) with $L = -d^2/dx^2$ and $f = 0$, whose solution clearly is $u = 0$ in (a,b). This is not restrictive since at every step the error: $u - u_1^{(k)}$ in Ω_1, $u - u_2^{(k)}$ in Ω_2, satisfies a problem like (18.5)-(18.6) with null forcing term.
Let $k = 1$; since $(u_1^{(1)})'' = 0$, $u_1^{(1)}(x)$ is a linear function; moreover, it vanishes at $x = a$ and takes the value $u_2^{(0)}$ at $x = \gamma_1$. As we know the value of $u_1^{(1)}$ at γ_2, we can solve the problem (18.6) which, in its turn, features a linear solution. Then we proceed in a similar manner. In Fig. 18.2 we show a few iterations: we clearly see that the method converges, moreover the convergence rate reduces as the length of the interval (γ_2, γ_1) gets smaller. ■

At each iteration the Schwarz iterative method (18.3)–(18.4) requires the solution of two subproblems with boundary conditions of the same kind as those of the original problem: indeed, by starting with a Dirichlet boundary-value problem in Ω we end up with two subproblems with Dirichlet conditions on the boundary of Ω_1 and Ω_2.
Should the differential problem (18.1) had been completed by a Neumann boundary condition on the whole boundary $\partial\Omega$, we would have been led to the solution of a mixed Dirichlet-Neumann boundary-value problem on either subdomain Ω_1 and Ω_2.

18.1.2 Dirichlet-Neumann method

Let us partition the domain Ω in two disjoint subdomains (as in Fig. 18.1): let then Ω_1 and Ω_2 be two subdomains providing a partition of Ω, i.e. $\overline{\Omega}_1 \cup \overline{\Omega}_2 = \overline{\Omega}$, $\overline{\Omega}_1 \cap \overline{\Omega}_2 = \Gamma$ and $\Omega_1 \cap \Omega_2 = \emptyset$. We denote by $\mathbf{n}_i$ the outward unit normal vector to Ω_i and will use the following notational convention: $\mathbf{n} = \mathbf{n}_1 = -\mathbf{n}_2$.

The following result holds (for its proof see [QV99]):

Theorem 18.1 (of equivalence). *The solution u of problem (18.1) is such that $u_{|\Omega_i} = u_i$ for $i = 1, 2$, where u_i is the solution to the problem*

$$\begin{cases} Lu_i = f & \text{in } \Omega_i, \\ u_i = 0 & \text{on } \partial\Omega_i \setminus \Gamma, \end{cases} \tag{18.8}$$

with interface conditions

$$u_1 = u_2 \tag{18.9}$$

and

$$\frac{\partial u_1}{\partial n_L} = \frac{\partial u_2}{\partial n_L} \tag{18.10}$$

on Γ, having denoted with $\partial/\partial n_L$ the conormal derivative (see (3.34)).

Thanks to this result we could split problem (18.1) by assigning the interface conditions (18.9)-(18.10) the role of "boundary conditions" for the two subproblems on the interface Γ. In particular, we can set up the following *Dirichlet-Neumann* (DN) iterative algorithm : given $u_2^{(0)}$ on Γ, for $k \geq 1$ solve the problems:

$$\begin{cases} Lu_1^{(k)} = f & \text{in } \Omega_1, \\ u_1^{(k)} = u_2^{(k-1)} & \text{on } \Gamma, \\ u_1^{(k)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma, \end{cases} \tag{18.11}$$

$$\begin{cases} Lu_2^{(k)} = f & \text{in } \Omega_2, \\ \dfrac{\partial u_2^{(k)}}{\partial n} = \dfrac{\partial u_1^{(k)}}{\partial n} & \text{on } \Gamma, \\ u_2^{(k)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma. \end{cases} \tag{18.12}$$

Condition (18.9) has generated a Dirichlet boundary condition on Γ for the subproblem in Ω_1 whereas (18.10) has generated a Neumann boundary condition on Γ for the subproblem in Ω_2.

Differently than Schwarz's method, the DN algorithm yields a Neumann boundary-value problem on the subdomain Ω_2. Theorem 18.1 guarantees that when the two sequences $\{u_1^{(k)}\}$ and $\{u_2^{(k)}\}$ converge, then their limit will be perforce the solution to

the exact problem (18.1). The DN algorithm is therefore *consistent*. Its convergence however is not always guaranteed, as we can see on the following simple example.

Example 18.2. Let $\Omega = (a,b)$, $\gamma \in (a,b)$, $L = -d^2/dx^2$ and $f = 0$. At every $k \geq 1$ the DN algorithm generates the two subproblems:

$$\begin{cases} -(u_1^{(k)})'' = 0, & a < x < \gamma, \\ u_1^{(k)} = 0, & x = a, \\ u_1^{(k)} = u_2^{(k-1)}, & x = \gamma, \end{cases} \tag{18.13}$$

$$\begin{cases} -(u_2^{(k)})'' = 0, & \gamma < x < b, \\ (u_2^{(k)})' = (u_1^{(k)})', & x = \gamma, \\ u_2^{(k)} = 0, & x = b. \end{cases} \tag{18.14}$$

Proceeding as done in Example 18.1, we can prove that the two sequences converge only if $\gamma > (a+b)/2$, as shown graphycally in Fig. 18.3.

■

In general, for a problem in arbitrary dimension $d > 1$, the measure of the "Dirichlet" subdomain Ω_1 must be larger than that of the "Neumann" one Ω_2 in order to guarantee the convergence of (18.11)-(18.12). This however yields a severe constraint to fulfill, especially if several subdomains will be used.
To overcome such limitation, a variant of the DN algorithm can be set up by replacing the Dirichlet condition $(18.11)_2$ in the first subdomain by

$$u_1^{(k)} = \theta u_2^{(k-1)} + (1-\theta) u_1^{(k-1)} \quad \text{on } \Gamma, \tag{18.15}$$

that is by introducing a *relaxation* which depends on a positive parameter θ. In such a way it is always possible to reduce the error between two subsequent iterates.

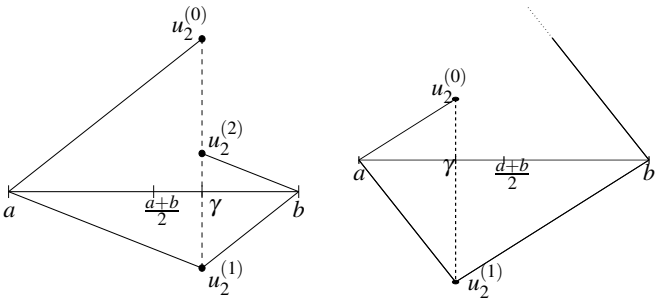

Fig. 18.3. Example of converging (left) and diverging (right) iterations for the DN method in 1D

In the case displayed in Fig. 18.3 we can easily verify that, by choosing

$$\theta_{opt} = -\frac{u_1^{(k-1)}}{u_2^{(k-1)} - u_1^{(k-1)}}, \tag{18.16}$$

the algorithm converges to the exact solution in a single iteration.

More in general, it can be proven that in any dimension $d \geq 1$, there exists a suitable value $\theta_{\max} < 1$ such that the DN algorithm converges for any possible choice of the relaxation parameter θ in the interval $(0, \theta_{\max})$.

18.1.3 Neumann-Neumann algorithm

Consider again a partition of Ω into two disjoint subdomains and denote by λ the (unknown) value of the solution u at their interface Γ. Consider the following iterative algorithm: for any given $\lambda^{(0)}$ on Γ, for $k \geq 0$ and $i = 1, 2$ solve the following problems:

$$\begin{cases} -\triangle u_i^{(k+1)} = f & \text{in } \Omega_i, \\ u_i^{(k+1)} = \lambda^{(k)} & \text{on } \Gamma, \\ u_i^{(k+1)} = 0 & \text{on } \partial\Omega_i \setminus \Gamma, \end{cases} \tag{18.17}$$

$$\begin{cases} -\triangle \psi_i^{(k+1)} = 0 & \text{in } \Omega_i, \\ \dfrac{\partial \psi_i^{(k+1)}}{\partial n} = \sigma_i \left(\dfrac{\partial u_1^{(k+1)}}{\partial n} - \dfrac{\partial u_2^{(k+1)}}{\partial n} \right) & \text{on } \Gamma, \\ \psi_i^{(k+1)} = 0 & \text{on } \partial\Omega_i \setminus \Gamma, \end{cases} \tag{18.18}$$

with

$$\lambda^{(k+1)} = \lambda^{(k)} - \theta \left(\sigma_1 \psi_{1|\Gamma}^{(k+1)} - \sigma_2 \psi_{2|\Gamma}^{(k+1)} \right), \tag{18.19}$$

where θ is a positive acceleration parameter, while σ_1 and σ_2 are two positive coefficients such that $\sigma_1 + \sigma_2 = 1$. This iterative algorithm is named *Neumann-Neumann* (NN). Note that in the first stage (18.17) we care about the continuity on Γ of the functions $u_1^{(k+1)}$ and $u_2^{(k+1)}$ but not that of their derivatives. The latter are addressed in the second stage (18.18), (18.19) by means of the correcting functions $\psi_1^{(k+1)}$ and $\psi_2^{(k+1)}$.

18.1.4 Robin-Robin algorithm

At last, we consider the following iterative algorithm, named *Robin-Robin* (RR). For every $k \geq 0$ solve the following problems:

$$\begin{cases} -\triangle u_1^{(k+1)} = f & \text{in } \Omega_1, \\ u_1^{(k+1)} = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ \dfrac{\partial u_1^{(k+1)}}{\partial n} + \gamma_1 u_1^{(k+1)} = \dfrac{\partial u_2^{(k)}}{\partial n} + \gamma_1 u_2^{(k)} & \text{on } \Gamma, \end{cases} \tag{18.20}$$

then

$$\begin{cases} -\triangle u_2^{(k+1)} = f & \text{in } \Omega_2, \\ u_2^{(k+1)} = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega, \\ \dfrac{\partial u_2^{(k+1)}}{\partial n} + \gamma_2 u_2^{(k+1)} = \dfrac{\partial u_1^{(k+1)}}{\partial n} + \gamma_2 u_1^{(k+1)} & \text{on } \Gamma, \end{cases} \tag{18.21}$$

where u_0 is assigned and γ_1, γ_2 are non-negative acceleration parameters that satisfy $\gamma_1 + \gamma_2 > 0$. Aiming at the algorithm parallelization, in (18.21) we could use $u_1^{(k)}$ instead of $u_1^{(k+1)}$, provided in such a case an initial value for u_1^0 is assigned as well.

18.2 Multi-domain formulation of Poisson problem and interface conditions

In this section, for the sake of exposition, we choose $L = -\triangle$ and consider the Poisson problem with homogeneous Dirichlet boundary conditions (3.13). Generalization to an arbitrary second order elliptic operator with different boundary conditions is in order.

In the case addressed in Sect. 18.1.2 of a domain partitioned into two disjoint subdomains, the equivalence Theorem 18.1 allows the following *multidomain formulation* of problem (18.1), in which $u_i = u\big|_{\Omega_i}$, $i = 1, 2$:

$$\begin{cases} -\triangle u_1 = f & \text{in } \Omega_1, \\ u_1 = 0 & \text{on } \partial\Omega_1 \setminus \Gamma, \\ -\triangle u_2 = f & \text{in } \Omega_2, \\ u_2 = 0 & \text{on } \partial\Omega_2 \setminus \Gamma, \\ u_1 = u_2 & \text{on } \Gamma, \\ \dfrac{\partial u_1}{\partial n} = \dfrac{\partial u_2}{\partial n} & \text{on } \Gamma. \end{cases} \tag{18.22}$$

18.2.1 The Steklov-Poincaré operator

We denote again by λ the unknown value of the solution u of problem (3.13) on the interface Γ, that is $\lambda = u_{|\Gamma}$. Should we know a priori the value λ on Γ, we could solve the following two independent boundary-value problems with Dirichlet condition on Γ $(i = 1, 2)$:

$$\begin{cases} -\triangle w_i = f & \text{in } \Omega_i\,, \\ w_i = 0 & \text{on } \partial\Omega_i \setminus \Gamma, \\ w_i = \lambda & \text{on } \Gamma. \end{cases} \tag{18.23}$$

With the aim of obtaining the value λ on Γ, let us split w_i as follows

$$w_i = w_i^* + u_i^0,$$

where w_i^* and u_i^0 represent the solutions of the following problems $(i = 1,2)$:

$$\begin{cases} -\triangle w_i^* = f & \text{in } \Omega_i\,, \\ w_i^* = 0 & \text{on } \partial\Omega_i \cap \partial\Omega, \\ w_i^* = 0 & \text{on } \Gamma, \end{cases} \tag{18.24}$$

and

$$\begin{cases} -\triangle u_i^0 = 0 & \text{in } \Omega_i\,, \\ u_i^0 = 0 & \text{on } \partial\Omega_i \cap \partial\Omega, \\ u_i^0 = \lambda & \text{on } \Gamma, \end{cases} \tag{18.25}$$

respectively. Note that the functions w_i^* depend solely on the source data f, while u_i^0 solely on the value λ on Γ, henceforth we can write $w_i^* = G_i f$ and $u_i^0 = H_i\lambda$. Both operators G_i and H_i are linear; H_i is the so-called harmonic extension operator of λ on the domain Ω_i.
By a formal comparison of problem (18.22) with problem (18.23), we infer that the equality

$$u_i = w_i^* + u_i^0\,,\ i = 1,2\,,$$

holds iff the condition $(18.22)_6$ on the normal derivatives on Γ is satisfied, that is iff

$$\frac{\partial w_1}{\partial n} = \frac{\partial w_2}{\partial n} \qquad \text{on } \Gamma.$$

By using the previously introduced notations the latter condition can be reformulated as

$$\frac{\partial}{\partial n}(G_1 f + H_1\lambda) = \frac{\partial}{\partial n}(G_2 f + H_2\lambda)$$

and therefore

$$\left(\frac{\partial H_1}{\partial n} - \frac{\partial H_2}{\partial n}\right)\lambda = \left(\frac{\partial G_2}{\partial n} - \frac{\partial G_1}{\partial n}\right) f \qquad \text{on } \Gamma.$$

In this way we have obtained an equation for the unknown λ on the interface Γ, named *Steklov-Poincaré equation*, that can be rewritten in compact form as

$$S\lambda = \chi \qquad \text{on } \Gamma. \tag{18.26}$$

S is the *Steklov-Poincaré* pseudo-differential operator; its formal definition is

$$S\mu = \frac{\partial}{\partial n} H_1\mu - \frac{\partial}{\partial n} H_2\mu = \sum_{i=1}^{2} \frac{\partial}{\partial n_i} H_i\mu = \sum_{i=1}^{2} S_i\mu, \tag{18.27}$$

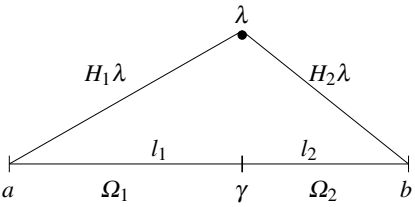

Fig. 18.4. Harmonic extensions in one dimension

while χ is a linear functional which depends on f

$$\chi = \frac{\partial}{\partial n} G_2 f - \frac{\partial}{\partial n} G_1 f = -\sum_{i=1}^{2} \frac{\partial}{\partial n_i} G_i f. \tag{18.28}$$

The operator

$$S_i : \mu \to S_i \mu = \frac{\partial}{\partial n_i} \left(H_i \mu \right) \Big|_{\Gamma}, \quad i = 1,2, \tag{18.29}$$

is called local Steklov-Poincaré operator. Note that S, S_1 and S_2 operate between the trace space

$$\Lambda = \{ \mu \mid \exists v \in V \ : \ \mu = v \big|_{\Gamma} \} \tag{18.30}$$

(that is $H_{00}^{1/2}(\Gamma)$, see [QV99]), and its dual Λ', whereas $\chi \in \Lambda'$.

Example 18.3. With the aim of providing a practical (elementary) example of operator S, let us consider a simple one-dimensional problem. Let $\Omega = (a,b) \subset \mathbb{R}$ as shown in Fig. 18.4 and $Lu = -u''$. By subdividing Ω in two disjoint subdomains, the interface Γ reduces to a single point $\gamma \in (a,b)$, and the Steklov-Poincaré operator S becomes

$$S\lambda = \left(\frac{dH_1}{dx} - \frac{dH_2}{dx} \right) \lambda = \left(\frac{1}{l_1} + \frac{1}{l_2} \right) \lambda,$$

with $l_1 = \gamma - a$ and $l_2 = b - \gamma$. ■

18.2.2 Equivalence between Dirichlet-Neumann and Richardson methods

The Dirichlet-Neumann (DN) method introduced in Sect. 18.1.2 can be reinterpreted as a (preconditioned) Richardson method for the solution of the Steklov-Poincaré interface equation. To check this statement, consider again, for the sake of simplicity, a domain Ω partitioned into two disjoint subdomains Ω_1 and Ω_2 with interface Γ. Then we re-write the DN algorithm (18.11), (18.12), (18.15) in the case of the operator $L = -\Delta$: for a given λ^0, for $k \geq 1$ solve:

$$\begin{cases} -\triangle u_1^{(k)} = f_1 & \text{in } \Omega_1, \\ u_1^{(k)} = \lambda^{(k-1)} & \text{on } \Gamma, \\ u_1^{(k)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma, \end{cases} \tag{18.31}$$

$$\begin{cases} -\triangle u_2^{(k)} = f_2 & \text{in } \Omega_2, \\ \dfrac{\partial u_2^{(k)}}{\partial n_2} = \dfrac{\partial u_1^{(k)}}{\partial n_2} & \text{on } \Gamma, \\ u_2^{(k)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma, \end{cases} \tag{18.32}$$

$$\lambda^{(k)} = \theta u_{2|\Gamma}^{(k)} + (1-\theta)\lambda^{(k-1)}. \tag{18.33}$$

The following result holds:

Theorem 18.2. *The Dirichlet-Neumann iterative algorithm (18.31)–(18.33) is equivalent to the preconditioned Richardson algorithm*

$$P_{DN}(\lambda^{(k)} - \lambda^{(k-1)}) = \theta(\chi - S\lambda^{(k-1)}). \tag{18.34}$$

The preconditioning operator is $P_{DN} = S_2$.

Proof. The solution $u_1^{(k)}$ of (18.31) can be written as

$$u_1^{(k)} = H_1\lambda^{(k-1)} + G_1 f_1. \tag{18.35}$$

Since $G_2 f_2$ satisfies the differential problem

$$\begin{cases} -\triangle(G_2 f_2) = f_2 & \text{in } \Omega_2, \\ G_2 f_2 = 0 & \text{on } \partial\Omega_2, \end{cases}$$

thanks to (18.32) the function $u_2^{(k)} - G_2 f_2$ satisfies the differential problem

$$\begin{cases} -\triangle(u_2^{(k)} - G_2 f_2) = 0 & \text{in } \Omega_2, \\ \dfrac{\partial}{\partial n_2}(u_2^{(k)} - G_2 f_2) = -\dfrac{\partial u_1^{(k)}}{\partial n} + \dfrac{\partial}{\partial n}(G_2 f_2) & \text{on } \Gamma, \\ u_2^{(k)} - G_2 f_2 = 0 & \text{on } \partial\Omega_2 \setminus \Gamma. \end{cases} \tag{18.36}$$

In particular $u_{2|\Gamma}^{(k)} = (u_2^{(k)} - G_2 f_2)_{|\Gamma}$. Since the operator S_i (18.29) maps a Dirichlet data to a Neumann data on Γ, its inverse S_i^{-1} transforms a Neumann data in a Dirichlet one on Γ.
Otherwise said, $S_2^{-1}\eta = w_{2|\Gamma}$, where w_2 is the solution of

$$\begin{cases} -\triangle w_2 = 0 & \text{in } \Omega_2, \\ \dfrac{\partial w_2}{\partial n} = \eta & \text{on } \Gamma, \\ w_2 = 0 & \text{on } \partial\Omega_2 \setminus \Gamma. \end{cases} \tag{18.37}$$

Setting now

$$\eta = -\frac{\partial u_1^{(k)}}{\partial n} + \frac{\partial}{\partial n}(G_2 f_2),$$

and comparing (18.36) with (18.37), we conclude that

$$u_2^{(k)}{}_{|\Gamma} = (u_2^{(k)} - G_2 f_2)_{|\Gamma} = S_2^{-1}\left(-\frac{\partial u_1^{(k)}}{\partial n} + \frac{\partial}{\partial n}(G_2 f_2)\right).$$

On the other hand, owing to (18.35) and to the definition (18.28) of χ, we obtain

$$\begin{aligned} u_2^{(k)}{}_{|\Gamma} &= S_2^{-1}\left(-\frac{\partial}{\partial n}(H_1\lambda^{(k-1)}) - \frac{\partial}{\partial n}(G_1 f_1) + \frac{\partial}{\partial n}(G_2 f_2)\right) \\ &= S_2^{-1}(-S_1\lambda^{(k-1)} + \chi). \end{aligned}$$

Using (18.33) we can therefore write

$$\lambda^{(k)} = \theta\left[S_2^{-1}(-S_1\lambda^{(k-1)} + \chi)\right] + (1-\theta)\lambda^{(k-1)},$$

that is

$$\lambda^{(k)} - \lambda^{(k-1)} = \theta\left[S_2^{-1}(-S_1\lambda^{(k-1)} + \chi) - \lambda^{(k-1)}\right].$$

Since $-S_1 = S_2 - S$, we finally obtain

$$\begin{aligned} \lambda^{(k)} - \lambda^{(k-1)} &= \theta\left[S_2^{-1}((S_2 - S)\lambda^{(k-1)} + \chi) - \lambda^{(k-1)}\right] \\ &= \theta S_2^{-1}(\chi - S\lambda^{(k-1)}), \end{aligned}$$

that is (18.34). The preconditioned *DN* operator is therefore $S_2^{-1}S = I + S_2^{-1}S_1$. ◇

Using an argument similar to that used for the proof of Theorem 18.2, also the Neumann-Neumann (NN) algorithm (18.17)–(18.19) can be interpreted as a preconditioned Richardson algorithm

$$P_{NN}(\lambda^{(k)} - \lambda^{(k-1)}) = \theta(\chi - S\lambda^{(k-1)}),$$

this time the preconditioner being $P_{NN} = (D_1 S_1^{-1} D_1 + D_2 S_2^{-1} D_2)^{-1}$ where D_i is a diagonal matrix whose entries are equal to σ_i. Note that the preconditioned operator becomes (if $D_i = I$) $S_2^{-1}S_1 + 2I + (S_2^{-1}S_1)^{-1}$.

Consider at last the Robin-Robin iterative algorithm (18.20)–(18.21). Denoting by $\mu_i^{(k)} \in \Lambda$ the approximation at step k of the trace of $u_i^{(k)}$ on the interface Γ, $i = 1, 2$, it can be proven that (18.20)–(18.21) is equivalent to the following alternating direction (ADI) algorithm:

$$\begin{aligned} (\gamma_1 i_\Lambda + S_1)\mu_1^{(k)} &= \chi + (\gamma_1 i_\Lambda + S_2)\mu_2^{(k-1)}, \\ (\gamma_2 i_\Lambda + S_2)\mu_2^{(k)} &= \chi + (\gamma_2 i_\Lambda + S_1)\mu_1^{(k-1)}, \end{aligned}$$

where $i_\Lambda : \Lambda \to \Lambda'$ here denotes the Riesz isomorphism between the Hilbert space Λ and its dual Λ' (see (2.5)).
Should, for a convenient choice of the two parameters γ_1 and γ_2, the algorithm converge to two limit functions μ_1 and μ_2, then $\mu_1 = \mu_2 = \lambda$, the latter function being the solution to the Steklov-Poincaré equation (18.26).
The RR preconditioner reads $P_{RR} = (\gamma_1 + \gamma_2)^{-1}(\gamma_1 i_\Lambda + S_1)(\gamma_2 i_\Lambda + S_2)$.

Remark 18.1. In the Dirichlet-Neumann algorithm, the value λ of the solution u at the interface Γ is the principal unknown. Once it has been determined, we can use it as Dirichlet data to recover the original solution in the whole domain. Alternatively, one could use the normal derivative $\eta = \frac{\partial u}{\partial n}$ on Γ as principal unknown (or, for a more general partial differential operator, the conormal derivative - or flux). By proceeding as above, we can show that η satisfies the new Steklov-Poincaré equation

$$(S_1^{-1} + S_2^{-1})\eta = T_1 f_1 + T_2 f_2 \quad \text{on}\,\Gamma \tag{18.38}$$

where for $i = 1, 2$, $T_i f_i$ is the solution of the following Neumann problem

$$\begin{cases} -\triangle(T_i f_i) = f_i & \text{in}\,\Omega_i, \\ \dfrac{\partial}{\partial n_i}(T_i f_i) = 0 & \text{on}\,\Gamma, \\ T_i f_i = 0 & \text{on}\,\partial\Omega \backslash \Gamma. \end{cases} \tag{18.39}$$

The so-called FETI algorithms (see Sect. 18.5.4) are examples of iterative algorithms designed for the solution of problems like (18.38). The FETI preconditioner is $P_{FETI} = S_1 + S_2$, hence the preconditioned FETI operator is $(S_1 + S_2)(S_1^{-1} + S_2^{-1})$.

•

18.3 Multidomain formulation of the finite element approximation of the Poisson problem

What seen thus far can be regarded as propedeutical to numerical solution of boundary-value problems. In this section we will see how the previous ideas can be reshaped in the framework of a numerical discretization method. Although we will only address the case of finite element discretization, this is however not restrictive. We refer, e.g., to [CHQZ07] and [TW05] for the case of spectral or spectral element discretizations and to [Woh01] for discretization based on DG and mortar methods.

Consider the Poisson problem (3.13), its weak formulation (3.18) and its Galerkin finite element approximation (4.40) on a triangulation $\mathcal{T}_h$. Recall that $V_h = \overset{\circ}{X}{}_h^r = \left\{ v_h \in X_h^r : v_h|_{\partial\Omega} = 0 \right\}$ is the space of finite element functions of degree r vanishing on $\partial\Omega$, whose basis is $\{\varphi_j\}_{j=1}^{N_h}$ (see Sect. 4.5.1).

For the finite element nodes in the domain Ω we consider the following partition: let $\{x_j^{(1)},\ 1 \le j \le N_1\}$ be the nodes located in the subdomain Ω_1, $\{x_j^{(2)},\ 1 \le j \le N_2\}$ those in Ω_2 and, finally, $\{x_j^{(\Gamma)},\ 1 \le j \le N_\Gamma\}$ those lying on the interface Γ. Let us split

the basis functions accordingly: $\varphi_j^{(1)}$ will denote those associated to the nodes $x_j^{(1)}$, $\varphi_j^{(2)}$ those associated with the nodes $x_j^{(2)}$, and $\varphi_j^{(\Gamma)}$ those associated with the nodes $x_j^{(\Gamma)}$ lying on the interface. This yields

$$\varphi_j^{(\alpha)}(x_j^{(\beta)}) = \delta_{ij}\delta_{\alpha\beta}\,, \quad 1 \le i \le N_\alpha, \quad \le j \le \mathbb{N}_\beta\,, \tag{18.40}$$

with $\alpha,\beta = 1,2,\Gamma$; δ_{ij} is the Kronecker symbol.
By letting v_h in (4.40) to coincide with a test function, (4.40) can be given the following equivalent formulation: find $u_h \in V_h$ such that

$$\begin{cases} a(u_h,\varphi_i^{(1)}) = F(\varphi_i^{(1)}) & \forall i = 1,\ldots,N_1\,, \\ a(u_h,\varphi_j^{(2)}) = F(\varphi_j^{(2)}) & \forall j = 1,\ldots,N_2\,, \\ a(u_h,\varphi_k^{(\Gamma)}) = F(\varphi_k^{(\Gamma)}) & \forall k = 1,\ldots,N_\Gamma\,, \end{cases} \tag{18.41}$$

having set $F(v) = \int_\Omega f v\,d\Omega$. Let now

$$a_i(v,w) = \int_{\Omega_i} \nabla v \cdot \nabla w\,d\Omega \qquad \forall v,w \in V,\ i = 1,2$$

be the restriction of the bilinear form $a(.,.)$ to the subdomain Ω_i and define $V_{i,h} = \{v \in H^1(\Omega_i)\,|\,v = 0 \text{ on } \partial\Omega_i \setminus \Gamma\}$ $(i = 1,2)$. Similarly we set $F_i(v) = \int_{\Omega_i} f v\,d\Omega$ and denote by $u_h^{(i)} = u_{h|_{\Omega_i}}$ the restriction of u_h to the subdomain Ω_i, with $i = 1,2$. Problem (18.41) can be rewritten in the equivalent form: find $u_h^{(1)} \in V_{1,h}$, $u_h^{(2)} \in V_{2,h}$ such that

$$\begin{cases} a_1(u_h^{(1)},\varphi_i^{(1)}) = F_1(\varphi_i^{(1)}) & \forall i = 1,\ldots,N_1, \\ a_2(u_h^{(2)},\varphi_j^{(2)}) = F_2(\varphi_j^{(2)}) & \forall j = 1,\ldots,N_2 \\ a_1(u_h^{(1)},\varphi_k^{(\Gamma)}|_{\Omega_1}) + a_2(u_h^{(2)},\varphi_k^{(\Gamma)}|_{\Omega_2}) & \\ = F_1(\varphi_k^{(\Gamma)}|_{\Omega_1}) + F_2(\varphi_k^{(\Gamma)}|_{\Omega_2}) & \forall k = 1,\ldots,N_\Gamma. \end{cases} \tag{18.42}$$

The interface continuity condition $(18.22)_5$ is automatically satisfied thanks to the continuity of the functions $u_h^{(i)}$. Moreover, equations $(18.42)_1$-$(18.42)_3$ correspond to the finite element discretization of equations $(18.22)_1$-$(18.22)_6$, respectively. In particular, the third of equations (18.42) must be regarded as the discrete counterpart of condition $(18.22)_6$ expressing the continuity of normal derivatives on Γ.
Let us expand the solution u_h with respect to the basis functions V_h

$$u_h(x) = \sum_{j=1}^{N_1} u_h(x_j^{(1)})\varphi_j^{(1)}(x) + \sum_{j=1}^{N_2} u_h(x_j^{(2)})\varphi_j^{(2)}(x) + \sum_{j=1}^{N_\Gamma} u_h(x_j^{(\Gamma)})\varphi_j^{(\Gamma)}(x). \tag{18.43}$$

From now on, the nodal values $u_h(x_j^{(\alpha)})$, for $\alpha = 1,2,\Gamma$ and $j = 1,\ldots,N_\alpha$, which are the expansion coefficients, will be indicated with the shorthand notation $u_j^{(\alpha)}$.

Using (18.43), we can rewrite (18.42) as follows:

$$\begin{cases} \displaystyle\sum_{j=1}^{N_1} u_j^{(1)} a_1(\varphi_j^{(1)}, \varphi_i^{(1)}) + \sum_{j=1}^{N_\Gamma} u_j^{(\Gamma)} a_1(\varphi_j^{(\Gamma)}, \varphi_i^{(1)}) = F_1(\varphi_i^{(1)}) \quad \forall i = 1, \ldots, N_1, \\ \displaystyle\sum_{j=1}^{N_2} u_j^{(2)} a_2(\varphi_j^{(2)}, \varphi_i^{(2)}) + \sum_{j=1}^{N_\Gamma} u_j^{(\Gamma)} a_2(\varphi_j^{(\Gamma)}, \varphi_i^{(2)}) = F_2(\varphi_i^{(2)}) \quad \forall i = 1, \ldots, N_2, \\ \displaystyle\sum_{j=1}^{N_\Gamma} u_j^{(\Gamma)} \left[a_1(\varphi_j^{(\Gamma)}, \varphi_i^{(\Gamma)}) + a_2(\varphi_j^{(\Gamma)}, \varphi_i^{(\Gamma)}) \right] \\ \quad + \displaystyle\sum_{j=1}^{N_1} u_j^{(1)} a_1(\varphi_j^{(1)}, \varphi_i^{(\Gamma)}) + \sum_{j=1}^{N_2} u_j^{(2)} a_2(\varphi_j^{(2)}, \varphi_i^{(\Gamma)}) \\ \quad = F_1(\varphi_i^{(\Gamma)}|_{\Omega_1}) + F_2(\varphi_i^{(\Gamma)}|_{\Omega_2}) \qquad \forall i = 1, \ldots, N_\Gamma. \end{cases} \tag{18.44}$$

Let us introduce the following arrays:

$$\begin{aligned} &(A_{11})_{ij} = a_1(\varphi_j^{(1)}, \varphi_i^{(1)}), && (A_{1\Gamma})_{ij} = a_1(\varphi_j^{(\Gamma)}, \varphi_i^{(1)}), \\ &(A_{22})_{ij} = a_2(\varphi_j^{(2)}, \varphi_i^{(2)}), && (A_{2\Gamma})_{ij} = a_2(\varphi_j^{(\Gamma)}, \varphi_i^{(2)}), \\ &\left(A^1_{\Gamma\Gamma}\right)_{ij} = a_1(\varphi_j^{(\Gamma)}, \varphi_i^{(\Gamma)}), && \left(A^2_{\Gamma\Gamma}\right)_{ij} = a_2(\varphi_j^{(\Gamma)}, \varphi_i^{(\Gamma)}), \\ &(A_{\Gamma 1})_{ij} = a_1(\varphi_j^{(1)}, \varphi_i^{(\Gamma)}), && (A_{\Gamma 2})_{ij} = a_2(\varphi_j^{(2)}, \varphi_i^{(\Gamma)}), \\ &(\mathbf{f}_1)_i = F_1(\varphi_i^{(1)}), && (\mathbf{f}_2)_i = F_2(\varphi_i^{(2)}), \\ &\left(\mathbf{f}_1^\Gamma\right)_i = F_1(\varphi_i^{(\Gamma)}), && \left(\mathbf{f}_2^\Gamma\right)_i = F_2(\varphi_i^{(\Gamma)}, \varphi_i^{(1)}), \end{aligned}$$

then set

$$\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \boldsymbol{\lambda})^T, \text{with } \mathbf{u}_1 = \left(u_j^{(1)}\right), \mathbf{u}_2 = \left(u_j^{(2)}\right) \text{ and } \boldsymbol{\lambda} = \left(u_j^{(\Gamma)}\right). \tag{18.45}$$

Problem (18.44) can be casted in the following algebraic form

$$\begin{cases} A_{11}\mathbf{u}_1 + A_{1\Gamma}\boldsymbol{\lambda} = \mathbf{f}_1, \\ A_{22}\mathbf{u}_2 + A_{2\Gamma}\boldsymbol{\lambda} = \mathbf{f}_2, \\ A_{\Gamma 1}\mathbf{u}_1 + A_{\Gamma 2}\mathbf{u}_2 + \left(A^{(1)}_{\Gamma\Gamma} + A^{(2)}_{\Gamma\Gamma}\right)\boldsymbol{\lambda} = \mathbf{f}_1^\Gamma + \mathbf{f}_2^\Gamma, \end{cases} \tag{18.46}$$

or, equivalently,

$$A\mathbf{u} = \mathbf{f}, \text{ that is } \begin{bmatrix} A_{11} & 0 & A_{1\Gamma} \\ 0 & A_{22} & A_{2\Gamma} \\ A_{\Gamma 1} & A_{\Gamma 2} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_\Gamma \end{bmatrix}, \tag{18.47}$$

having set $A_{\Gamma\Gamma} = \left(A^{(1)}_{\Gamma\Gamma} + A^{(2)}_{\Gamma\Gamma}\right)$ and $\mathbf{f}_\Gamma = \mathbf{f}_1^\Gamma + \mathbf{f}_2^\Gamma$. (18.47) is nothing but a blockwise representation of the finite element system (4.46), the blocks being determined by the partition (18.45) of the vector of unknowns.

More precisely, the first and second equations of (18.46) are discretizations of the given Poisson problems in Ω_1 and Ω_2, respectively for the interior values $\mathbf{u}_1$ and $\mathbf{u}_2$, with Dirichlet data vanishing on $\partial\Omega_i \setminus \Gamma$ and equal to the common value $\boldsymbol{\lambda}$ on Γ.
Alternatively, by setting (from the third equation of (18.46))

$$A_{\Gamma_1}\mathbf{u}_1 + A^{(1)}_{\Gamma\Gamma}\boldsymbol{\lambda} - \mathbf{f}^1_\Gamma = -(A_{\Gamma_2}\mathbf{u}_2 + A^{(2)}_{\Gamma\Gamma}\boldsymbol{\lambda} - \mathbf{f}^2_\Gamma) \equiv \boldsymbol{\eta}, \tag{18.48}$$

the first and third equations of (18.46) provide a discretization of the Poisson problem in Ω_1 with vanishing Dirichlet data on $\partial\Omega_1 \setminus \Gamma$ and with Neumann data $\boldsymbol{\eta}$ on Γ.

Similar considerations apply to the second and third equations of (18.46): they represent the discretization of a Poisson problem in Ω_2 with zero Dirichlet data in $\partial\Omega_2 \setminus \Gamma$ and Neumann data equal to $\boldsymbol{\eta}$ on Γ.

18.3.1 The Schur complement

Consider now the Steklov-Poincaré interface equation (18.26) and look for its finite element counterpart. Since λ represents the unknown value of u on Γ, its finite element correspondent is the vector $\boldsymbol{\lambda}$ of the values of u_h at the interface nodes.

By gaussian elimination operated on system (18.47), we can obtain a new reduced system on the sole unknown $\boldsymbol{\lambda}$.

Matrices A_{11} and A_{22} are invertible since they are associated with two homogeneous Dirichlet boundary-value problems for the Laplace operator, hence

$$\mathbf{u}_1 = A_{11}^{-1}(\mathbf{f}_1 - A_{1\Gamma}\boldsymbol{\lambda}) \quad \text{and} \quad \mathbf{u}_2 = A_{22}^{-1}(\mathbf{f}_2 - A_{2\Gamma}\boldsymbol{\lambda}). \tag{18.49}$$

From the third equation in (18.46), we obtain

$$\begin{aligned}&\left[\left(A^{(1)}_{\Gamma\Gamma} - A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma}\right) + \left(A^{(2)}_{\Gamma\Gamma} - A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma}\right)\right]\boldsymbol{\lambda} \\ &= \mathbf{f}_\Gamma - A_{\Gamma 1}A_{11}^{-1}\mathbf{f}_1 - A_{\Gamma 2}A_{22}^{-1}\mathbf{f}_2 = (\mathbf{f}^{(1)}_\Gamma - A_{\Gamma_1}A_{11}^{-1}\mathbf{f}_1) + (\mathbf{f}^{(2)}_\Gamma - A_{\Gamma_2}A_{22}^{-1}\mathbf{f}_2).\end{aligned} \tag{18.50}$$

Using the following definitions:

$$\Sigma = \Sigma_1 + \Sigma_2, \qquad \Sigma_i = A^{(i)}_{\Gamma\Gamma} - A_{\Gamma i}A_{ii}^{-1}A_{i\Gamma}, \qquad i = 1,2, \tag{18.51}$$

and

$$\boldsymbol{\chi}_\Gamma = \boldsymbol{\chi}^{(1)}_\Gamma + \boldsymbol{\chi}^{(2)}_\Gamma, \quad \boldsymbol{\chi}^{(i)}_\Gamma = \mathbf{f}^{(i)}_\Gamma - A_{\Gamma_i}A_{ii}\mathbf{f}_i, \tag{18.52}$$

(18.50) becomes

$$\Sigma\boldsymbol{\lambda} = \boldsymbol{\chi}_\Gamma. \tag{18.53}$$

Since Σ and $\boldsymbol{\chi}_\Gamma$ approximate S and $\boldsymbol{\chi}$, respectively, (18.53) can be considered as a finite element approximation to the Steklov-Poincaré equation (18.26). Matrix Σ is the so-called *Schur complement* of A with respect to $\mathbf{u}_1$ and $\mathbf{u}_2$, whereas matrices Σ_i are the Schur complements related to the subdomains Ω_i $(i = 1,2)$.
Once system (18.53) is solved w.r.t the unknown $\boldsymbol{\lambda}$, by virtue of (18.49) we can compute $\mathbf{u}_1$ and $\mathbf{u}_2$. This computation amounts to solve numerically two Poisson problems

on the two subdomains Ω_1 and Ω_2, with Dirichlet boundary conditions $u_h^{(i)}{}_{|\Gamma} = \lambda_h$ ($i = 1, 2$) on the interface Γ.

The Schur complement Σ inherits some of the properties of its generating matrix A, as stated by the following result:

Lemma 18.1. *Matrix Σ satisfies the following properties:*

1. *if A is singular, so is Σ;*
2. *if A (respectively, A_{ii}) is symmetric, then Σ (respectively, Σ_i) is symmetric too;*
3. *if A is positive definite, so is Σ.*

Recall that the condition number of the finite element stiffness matrix A satisfies $K_2(A) \simeq Ch^{-2}$ (see (4.50)). As of Σ, it can be proven that

$$K_2(\Sigma) \simeq Ch^{-1}. \tag{18.54}$$

In the specific case under consideration, A (and therefore Σ, thanks to Lemma 18.1) is symmetric and positive definite. It is therefore convenient to use the conjugate gradient method (with a suitable preconditioner) for the solution of system (18.53). At every iteration, the computation of the residue will involve the finite element solution of two independent Dirichlet boundary-value problems on the subdomains Ω_i.

By employing a similar procedure we can derive instead of (18.53) an interface equation for the flux $\boldsymbol{\eta}$ introduced in (18.48). From (18.47) and (18.48) we derive

$$\begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma}^{(1)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma^{(1)} + \boldsymbol{\eta} \end{bmatrix}. \tag{18.55}$$

By eliminating $\mathbf{u}_1$ from the first row and replacing it in the second one we obtain

$$\Sigma_1 \boldsymbol{\lambda} = \boldsymbol{\chi}_\Gamma^{(1)} + \boldsymbol{\eta}, \text{ that is } \boldsymbol{\lambda} = \Sigma_1^{-1}(\boldsymbol{\chi}_\Gamma^{(1)} + \boldsymbol{\eta}). \tag{18.56}$$

Proceeding in a similar way we obtain

$$\Sigma_2 \boldsymbol{\lambda} = \boldsymbol{\chi}_\Gamma^{(2)} - \boldsymbol{\eta}, \text{ that is } \boldsymbol{\lambda} = \Sigma_2^{-1}(\boldsymbol{\chi}_\Gamma^{(2)} - \boldsymbol{\eta}). \tag{18.57}$$

By equating the last two equations (whose common value is $\boldsymbol{\lambda}$) we finally obtain the Schur-complement equation for the flux $\boldsymbol{\eta}$:

$$T\boldsymbol{\eta} = \psi_\Gamma, \quad \text{with } T = \Sigma_1^{-1} + \Sigma_2^{-1}, \psi_\Gamma = \Sigma_2^{-1}\boldsymbol{\chi}_\Gamma^{(2)} - \Sigma_1^{-1}\boldsymbol{\chi}_\Gamma^{(1)}. \tag{18.58}$$

This algebraic equation can be regarded as a direct discretization of the Steklov-Poincaré problem for the flux (18.38).

18.3.2 The discrete Steklov-Poincaré operator

In this section we will find the discrete operator associated with the Schur complement. With this aim, besides the space $V_{i,h}$ previously introduced, we will need the one $V_{i,h}^0$ generated by the functions $\{\varphi_j^{(i)}\}$ exclusively associated to the internal nodes of the subdomain Ω_i, and the space Λ_h generated by the set of functions $\{\varphi_j^{(\Gamma)}{}_{|_\Gamma}\}$. We have $\Lambda_h = \{\mu_h \mid \exists v_h \in V_h \ : \ v_h\big|_\Gamma = \mu_h\}$, whence Λ_h represents a finite element subspace of the trace functions space Λ introduced in (18.30).
Consider now the following problem: find $H_{i,h}\eta_h \in V_{i,h}$, with $H_{i,h}\eta_h = \eta_h$ on Γ, such that

$$\int_{\Omega_i} \nabla(H_{i,h}\eta_h)\cdot\nabla v_h \, d\Omega_i = 0 \qquad \forall v_h \in V_{i,h}^0. \tag{18.59}$$

Clearly, $H_{i,h}\eta_h$ represents a finite element approximation of the harmonic extension $H_i\eta_h$, and the operator $H_{i,h} : \eta_h \to H_{i,h}\eta_h$ can be regarded as an approximation of H_i. By expanding $H_{i,h}\eta_h$ in terms of the basis functions

$$H_{i,h}\eta_h = \sum_{j=1}^{N_i} u_j^{(i)}\varphi_j^{(i)} + \sum_{k=1}^{N_\Gamma} \eta_k \varphi_k^{(\Gamma)}{}_{|_{\Omega_i}},$$

we can rewrite (18.59) in matrix form

$$A_{ii}\mathbf{u}^{(i)} = -A_{i\Gamma}\boldsymbol{\eta}. \tag{18.60}$$

The following result, called *the uniform discrete extension theorem*, holds:

Theorem 18.3. *There exist two constants $\hat{C}_1, \hat{C}_2 > 0$, independent of h, such t hat*

$$\hat{C}_1\|\eta_h\|_\Lambda \le \|H_{i,h}\eta_h\|_{H^1(\Omega_i)} \le \hat{C}_2\|\eta_h\|_\Lambda \qquad \forall \eta_h \in \Lambda_h \ \ i = 1,2. \tag{18.61}$$

Consequently, there exist two constants $K_1, K_2 > 0$, independent of h, such that

$$K_1\|H_{1,h}\eta_h\|_{H^1(\Omega_1)} \le \|H_{2,h}\eta_h\|_{H^1(\Omega_2)} \le K_2\|H_{1,h}\eta_h\|_{H^1(\Omega_1)} \quad \forall \eta_h \in \Lambda_h. \tag{18.62}$$

For the proof see, e.g., [QV99].

Now for $i = 1,2$ the (local) discrete Steklov-Poincaré operator is defined as follows: $S_{i,h} : \Lambda_h \to \Lambda_h'$,

$$\langle S_{i,h}\eta_h, \mu_h\rangle = \int_{\Omega_i} \nabla(H_{i,h}\eta_h)\cdot\nabla(H_{i,h}\mu_h)d\Omega_i \quad \forall \eta_h, \mu_h \in \Lambda_h, \tag{18.63}$$

then we define the (global) discrete Steklov-Poincaré operator as $S_h = S_{1,h} + S_{2,h}$.

Lemma 18.2. *The local discrete Steklov-Poincaré operator can be expressed in terms of the local Schur complement as*

$$\langle S_{i,h}\eta_h, \mu_h\rangle = \boldsymbol{\mu}^T \Sigma_i \boldsymbol{\eta} \quad \forall \eta_h, \mu_h \in \Lambda_h \,, i = 1,2, \tag{18.64}$$

where

$$\eta_h = \sum_{k=1}^{N_\Gamma} \eta_k \varphi_k^{(\Gamma)}{}_{|\Gamma}, \quad \mu_h = \sum_{k=1}^{N_\Gamma} \mu_k \varphi_k^{(\Gamma)}{}_{|\Gamma}$$

and

$$\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{N_\Gamma})^T, \quad \boldsymbol{\mu} = (\mu_1, \ldots, \mu_{N_\Gamma})^T.$$

Therefore the global discrete Steklov-Poincaré operator $S_h = S_{1,h} + S_{2,h}$ satisfies the relation

$$\langle S_h \eta_h, \mu_h\rangle = \boldsymbol{\mu}^T \Sigma \boldsymbol{\eta} \quad \forall \eta_h, \mu_h \in \Lambda_h. \tag{18.65}$$

Proof. For $i = 1,2$ we have

$$\begin{aligned}
\langle S_{i,h}\eta_h, \mu_h\rangle &= a_i(H_{i,h}\eta_h, H_{i,h}\mu_h) \\
&= a_i\Big(\sum_{j=1}^{N_\Gamma} u_j \varphi_j^{(i)} + \sum_{k=1}^{N_\Gamma} \eta_k \varphi_k^{(\Gamma)}{}_{|\Omega_i}, \sum_{l=1}^{N_\Gamma} w_l \varphi_l^{(i)} + \sum_{m=1}^{N_\Gamma} \mu_m \varphi_m^{(\Gamma)}{}_{|\Omega_i}\Big) \\
&= \sum_{j,l=1}^{N_\Gamma} w_l a_i(\varphi_j^{(i)}, \varphi_l^{(i)}) u_j + \sum_{j,m=1}^{N_\Gamma} \mu_m a_i(\varphi_j^{(i)}, \varphi_m^{(\Gamma)}{}_{|\Omega_i}) u_j \\
&\quad + \sum_{k,l=1}^{N_\Gamma} w_l a_i(\varphi_k^{(\Gamma)}{}_{|\Omega_i}, \varphi_l^{(i)}) \eta_k + \sum_{k,m=1}^{N_\Gamma} \mu_m a_i(\varphi_k^{(\Gamma)}{}_{|\Omega_i}, \varphi_m^{(\Gamma)}{}_{|\Omega_i}) \eta_k \\
&= \mathbf{w}^T A_{ii}\mathbf{u} + \boldsymbol{\mu}^T A_{\Gamma i}\mathbf{u} + \mathbf{w}^T A_{i\Gamma}\boldsymbol{\eta} + \boldsymbol{\mu}^T A_{\Gamma\Gamma}^{(i)}\boldsymbol{\eta}.
\end{aligned}$$

Thanks to (18.60) we obtain

$$\begin{aligned}
\langle S_{i,h}\eta_h, \mu_h\rangle &= -\mathbf{w}^T A_{i\Gamma}\boldsymbol{\eta} - \boldsymbol{\mu}^T A_{\Gamma i} A_{ii}^{-1} A_{i\Gamma}\boldsymbol{\eta} + \mathbf{w}^T A_{i\Gamma}\boldsymbol{\eta} + \boldsymbol{\mu}^T A_{\Gamma\Gamma}^{(i)}\boldsymbol{\eta} \\
&= \boldsymbol{\mu}^T \left(A_{\Gamma\Gamma}^{(i)} - A_{\Gamma i} A_{ii}^{-1} A_{i\Gamma}\right)\boldsymbol{\eta} \\
&= \boldsymbol{\mu}^T \Sigma_i \boldsymbol{\eta}.
\end{aligned}$$

◇

From Theorem 18.3 and thanks to the representation (18.63), we deduce that there exist two constants $\hat{K}_1, \hat{K}_2 > 0$, independent of h, such that

$$\hat{K}_1 \langle S_{1,h}\mu_h, \mu_h\rangle \;\le\; \langle S_{2,h}\mu_h, \mu_h\rangle \;\le\; \hat{K}_2 \langle S_{1,h}\mu_h, \mu_h\rangle \qquad \forall \mu_h \in \Lambda_h. \tag{18.66}$$

Thanks to (18.64) we can infer that there exist two constants $\tilde{K}_1, \tilde{K}_2 > 0$, independent of h, such that

$$\tilde{K}_1 \left(\boldsymbol{\mu}^T \Sigma_1 \boldsymbol{\mu}\right) \leq \boldsymbol{\mu}^T \Sigma_2 \boldsymbol{\mu} \leq \tilde{K}_2 \left(\boldsymbol{\mu}^T \Sigma_1 \boldsymbol{\mu}\right) \qquad \forall \boldsymbol{\mu} \in \mathbb{R}^{N_\Gamma}. \tag{18.67}$$

This amounts to say that the two matrices Σ_1 and Σ_2 are spectrally equivalent, that is their spectral condition number features the same asymptotic behaviour w.r.t h. Henceforth, both Σ_1 and Σ_2 provide an optimal preconditioner of the Schur complement Σ, that is there exists a constant C, independent of h, such that

$$K_2(\Sigma_i^{-1}\Sigma) \leq C, \quad i = 1,2. \tag{18.68}$$

As we will see in Sect. 18.3.3, this property allows us to prove that the discrete version of the Dirichlet-Neumann algorithm converges with a rate independent of h. A similar result holds for the discrete Neumann-Neumann and Robin-Robin algorithms.

18.3.3 Equivalence between the Dirichlet-Neumann algorithm and a preconditioned Richardson algorithm in the discrete case

Let us now prove the analogue of the equivalence theorem 18.2 in the algebraic case. The finite element approximation of the Dirichlet problem (18.31) has the following algebraic form

$$A_{11}\mathbf{u}_1^{(k)} = \mathbf{f}_1 - A_{1\Gamma}\boldsymbol{\lambda}^{(k-1)}, \tag{18.69}$$

whereas that of the Neumann problem (18.32) reads

$$\begin{bmatrix} A_{22} & A_{2\Gamma} \\ A_{\Gamma 2} & A_{\Gamma\Gamma}^{(2)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_2^{(k)} \\ \boldsymbol{\lambda}^{(k-1/2)} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_2 \\ \mathbf{f}_\Gamma - A_{\Gamma 1}\mathbf{u}_1^{(k)} - A_{\Gamma\Gamma}^{(1)}\boldsymbol{\lambda}^{(k-1)} \end{bmatrix}. \tag{18.70}$$

In its turn, (18.33) becomes

$$\boldsymbol{\lambda}^{(k)} = \theta\boldsymbol{\lambda}^{(k-1/2)} + (1-\theta)\boldsymbol{\lambda}^{(k-1)}. \tag{18.71}$$

By eliminating $\mathbf{u}_2^{(k)}$ from (18.70) we obtain

$$\left(A_{\Gamma\Gamma}^{(2)} - A_{\Gamma 2}A_{22}^{-1}A_{2\Gamma}\right)\boldsymbol{\lambda}^{(k-1/2)} = \mathbf{f}_\Gamma - A_{\Gamma 1}\mathbf{u}_1^{(k)} - A_{\Gamma\Gamma}^{(1)}\boldsymbol{\lambda}^{(k-1)} - A_{\Gamma 2}A_{22}^{-1}\mathbf{f}_2.$$

By the definition (18.51) of Σ_2 and by (18.69), one has

$$\Sigma_2\boldsymbol{\lambda}^{(k-1/2)} = \mathbf{f}_\Gamma - A_{\Gamma 1}A_{11}^{-1}\mathbf{f}_1 - A_{\Gamma 2}A_{22}^{-1}\mathbf{f}_2 - \left(A_{\Gamma\Gamma}^{(1)} - A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma}\right)\boldsymbol{\lambda}^{(k-1)},$$

that is, owing to the definition (18.51) of Σ_1 and to (18.52),

$$\boldsymbol{\lambda}^{(k-1/2)} = \Sigma_2^{-1}\left(\chi_\Gamma - \Sigma_1\boldsymbol{\lambda}^{(k-1)}\right).$$

Now, by virtue of (18.71) we deduce

$$\boldsymbol{\lambda}^{(k)} = \theta\Sigma_2^{-1}\left(\chi_\Gamma - \Sigma_1\boldsymbol{\lambda}^{(k-1)}\right) + (1-\theta)\boldsymbol{\lambda}^{(k-1)},$$

that is, since $-\Sigma_1 = -\Sigma + \Sigma_2$,

$$\boldsymbol{\lambda}^{(k)} = \theta \Sigma_2^{-1} \left(\boldsymbol{\chi}_\Gamma - \Sigma \boldsymbol{\lambda}^{(k-1)} + \Sigma_2 \boldsymbol{\lambda}^{(k-1)} \right) + (1-\theta) \boldsymbol{\lambda}^{(k-1)}$$

whence

$$\Sigma_2 (\boldsymbol{\lambda}^{(k)} - \boldsymbol{\lambda}^{(k-1)}) = \theta (\boldsymbol{\chi}_\Gamma - \Sigma \boldsymbol{\lambda}^{(k-1)}).$$

The latter is nothing but a Richardson iteration on the system (18.53) using the local Schur complement Σ_2 as preconditioner.

Remark 18.2. The Richardson preconditioner induced by the Dirichlet-Neumann algorithm is in fact the local Schur complement associated to that subdomain on which we solve a Neumann problem. So, in the so-called *Neumann-Dirichlet* algorithm, in which at every iteration we solve a Dirichlet problem in Ω_2 and a Neumann one in Ω_1, the preconditioner of the associated Richardson algorithm would be Σ_1 and not Σ_2. •

Remark 18.3. An analogous result can be proven for the discrete version of the Neumann-Neumann algorithm introduced in Sect. 18.1.3. Precisely, the Neumann-Neumann algorithm is equivalent to the Richardson algorithm applied to system (18.53) with a preconditioner whose inverse is given by $P_h^{-1} = \sigma_1 \Sigma_1^{-1} + \sigma_2 \Sigma_2^{-1}$, σ_1 and σ_2 being the coefficients used for the (discrete) interface equation which corresponds to (18.19). Moreover we can prove that there exists a constant $C > 0$, independent of h, such that

$$K_2((\sigma_1 \Sigma_1^{-1} + \sigma_2 \Sigma_2^{-1})\Sigma) \leq C\,.$$

Proceeding in a similar way we can show that the discrete version of the Robin-Robin algorithm (18.20)-(18.21) is also equivalent to a Richardson algorithm for (18.53), using this time as preconditioner the matrix $(\gamma_1 + \gamma_2)^{-1}(\gamma_1 I + \Sigma_1)(\gamma_2 I + \Sigma_2)$. •

Let us recall that a matrix P_h is an optimal preconditioner for Σ if the condition number of $P_h^{-1}\Sigma$ is bounded uniformely w.r.t the dimension N of the matrix Σ (and therefore from h in the case in which Σ arises from a finite element discretization).

We can therefore summarize by saying that for the solution of system $\Sigma \boldsymbol{\lambda} = \boldsymbol{\chi}_\Gamma$, we can make use of the following preconditioners, all of them being optimal:

$$P_h = \begin{cases} \Sigma_2 & \text{for the Dirichlet-Neumann algorithm,} \\ \Sigma_1 & \text{for the Neumann-Dirichlet algorithm,} \\ (\sigma_1 \Sigma_1^{-1} + \sigma_2 \Sigma_2^{-1})^{-1} & \text{for the Neumann-Neumann algorithm,} \\ (\gamma_1 + \gamma_2)^{-1}(\gamma_1 I + \Sigma_1)(\gamma_2 I + \Sigma_2) & \text{for the Robin-Robin algorithm.} \end{cases} \tag{18.72}$$

When solving the flux equation (18.58), the FETI preconditioner reads $P_h = (\Sigma_1 + \Sigma_2)^{-1}$, yelding the preconditioned matrix $(\Sigma_1 + \Sigma_2)(\Sigma_1^{-1} + \Sigma_2^{-1})$. For all these preconditioners, optimality follows from the spectral equivalence (18.67), hence $K_2(P_h^{-1}\Sigma)$ is bounded independently of h.

From the convergence theory of Richardson method we know that if both Σ and P_h are symmetric and positive definite, one has $\|\boldsymbol{\lambda}^n - \boldsymbol{\lambda}\|_\Sigma \leq \rho^n \|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}\|_\Sigma$, $n \geq 0$, being $\|\boldsymbol{v}\|_\Sigma = (\boldsymbol{v}^T \Sigma \boldsymbol{v})^{1/2}$. The optimal convergence rate is given by

$$\rho = \frac{K_2(P_h^{-1}\Sigma) - 1}{K_2(P_h^{-1}\Sigma) + 1},$$

and is therefore independent of h.

18.4 Generalization to the case of many subdomains

To generalize the previous DD algorithms to the case in which the domain Ω is partitioned into an arbitrary number $M > 2$ of subdomains we proceed as follows.
Let Ω_i, $i = 1, \ldots, M$, denote a family of disjoint subdomains such that $\cup \overline{\Omega_i} = \overline{\Omega}$, and denote $\Gamma_i = \partial\Omega_i \setminus \partial\Omega$ and $\Gamma = \cup \Gamma_i$ (the skeleton).

Let us consider the Poisson problem (3.13). In the current case the equivalence Theorem 18.1 generalizes as follows:

$$\begin{cases} -\triangle u_i = f & \text{in } \Omega_i, \\ u_i = u_k & \text{on } \Gamma_{ik}, \ \forall k \in \mathcal{A}(i), \\ \dfrac{\partial u_i}{\partial n_i} = \dfrac{\partial u_k}{\partial n_i} & \text{on } \Gamma_{ik}, \ \forall k \in \mathcal{A}(i), \\ u_i = 0 & \text{on } \partial\Omega_i \cap \partial\Omega, \end{cases} \tag{18.73}$$

for $i = 1, \ldots, M$, being $\Gamma_{ik} = \partial\Omega_i \cap \partial\Omega_k \neq \emptyset$, $\mathcal{A}(i)$ the set of indices k such that Ω_k is adjacent to Ω_i; as usual, $\mathbf{n}_i$ denotes the outward unit normal vetor to Ω_i.

Assume now that (3.13) has been approximated by the finite element method. Following the ideas presented in Sect. 18.3 and denoting by $\mathbf{u} = (\mathbf{u}_I, \mathbf{u}_\Gamma)^T$ the vector of unknowns split in two subvectors, the one ($\mathbf{u}_I$) related with the internal nodes, and that ($\mathbf{u}_\Gamma$) related with the nodes lying on the skeleton Γ, the finite element algebraic system can be reformulated in blockwise form as follows

$$\begin{bmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix} \begin{bmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I \\ \mathbf{f}_\Gamma \end{bmatrix}, \tag{18.74}$$

being $A_{\Gamma I} = A_{I\Gamma}^T$. Matrix $A_{I\Gamma}$ is banded, while A_{II} has the block diagonal form

$$A_{II} = \begin{bmatrix} A_{\Omega_1,\Omega_1} & 0 & \ldots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & A_{\Omega_M,\Omega_M} \end{bmatrix}. \tag{18.75}$$

We are using the following notations:

$$
\begin{aligned}
&(A_{\Omega_i,\Omega_i})_{lj} = a_i(\varphi_j,\varphi_l), \quad 1 \le l,j \le N_i,\\
&(A^{(i)}_{\Gamma\Gamma})_{sr} = a_i(\psi_r,\psi_s), \quad 1 \le r,s \le N_{\Gamma_i},\\
&(A_{\Omega_i,\Gamma})_{lr} = a_i(\psi_r,\varphi_l), \quad 1 \le r \le N_{\Gamma_i}, \qquad 1 \le l \le N_i,
\end{aligned}
$$

where N_i is the number of nodes internal to Ω_i, N_{Γ_i} that of the nodes sitting on the interface Γ_i, φ_j and ψ_r the basis functions associated with the internal and interface nodes, respectively.

Let us remark that on every subdomain Ω_i the matrix

$$
A_i = \begin{bmatrix} A_{\Omega_i,\Omega_i} & A_{\Omega_i,\Gamma} \\ A_{\Gamma,\Omega_i} & A^{(i)}_{\Gamma\Gamma} \end{bmatrix} \tag{18.76}
$$

represents the local finite element stiffness matrix associated to a Neumann problem on Ω_i. Since A_{II} is non-singular, from (18.74) we can formally derive

$$
\mathbf{u}_I = A_{II}^{-1}(\mathbf{f}_I - A_{I\Gamma}\mathbf{u}_\Gamma). \tag{18.77}
$$

By eliminating the unknown $\mathbf{u}_I$ from system (18.74), it follows

$$
A_{\Gamma\Gamma}\mathbf{u}_\Gamma = \mathbf{f}_\Gamma - A_{\Gamma I}A_{II}^{-1}(\mathbf{f}_I - A_{I\Gamma}\mathbf{u}_\Gamma),
$$

that is

$$
\begin{pmatrix} A_{II} & A_{I\Gamma} \\ 0 & \Sigma \end{pmatrix} \begin{pmatrix} u_I \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{f}_I \\ \chi_\Gamma \end{pmatrix} \tag{18.78}
$$

having set

$$
\Sigma = A_{\Gamma\Gamma} - A_{\Gamma I}A_{II}^{-1}A_{I\Gamma} \ \text{ and } \ \chi_\Gamma = \mathbf{f}_\Gamma - A_{\Gamma I}A_{II}^{-1}\mathbf{f}_I.
$$

Denoting, as usual, $\boldsymbol{\lambda} = \mathbf{u}_\Gamma$, (18.78) yields

$$
\Sigma\boldsymbol{\lambda} = \chi_\Gamma. \tag{18.79}
$$

This is the Schur complement system in the multidomain case. It can be regarded as a finite element approximation of the interface Steklov-Poincaré problem in the case of M subdomains.

The local Schur complements are defined as

$$
\Sigma_i = A^{(i)}_{\Gamma\Gamma} - A_{\Gamma,\Omega_i}A^{-1}_{\Omega_i,\Omega_i}A_{\Omega_i,\Gamma}, \quad i = 1,\dots M, \tag{18.80}
$$

hence

$$
\Sigma = R^T_{\Gamma_1}\Sigma_1 R_{\Gamma_1} + \dots + R^T_{\Gamma_M}\Sigma_M R_{\Gamma_M} \tag{18.81}
$$

where R_{Γ_i} is a restriction operator, that is a rectangular matrix of zeros and ones that map values on Γ onto those on Γ_i, $i = 1,\dots,M$. Note that the r.h.s. of (18.79) can be

written as a sum of local contributions,

$$\chi_\Gamma = \sum_{i=1}^{M} R_{\Gamma_i}^T \left(\mathbf{f}_\Gamma^{(i)} - A_{\Gamma,\Omega_i} A_{\Omega_i,\Omega_i}^{-1} \mathbf{f}_I^{(i)} \right) \tag{18.82}$$

A general algorithm to solve the finite element Poisson problem in Ω could be formulated as follows:

1. compute the solution of (18.79) to obtain the value of $\boldsymbol{\lambda}$ on the skeleton Γ;
2. then solve (18.77); since A_{II} is block-diagonal, this step yields the solution of M independent subproblems of reduced dimension, $A_{\Omega_i,\Omega_i}\mathbf{u}_I^i = \mathbf{g}^i$, $i = 1,\ldots,M$, which can therefore be carried out in parallel.

About the condition number of Σ, the following estimate can be proven: there exists a constant $C > 0$, independent of h and H_{min}, H_{max}, such that

$$K_2(\Sigma) \leq C \frac{H_{max}}{h H_{min}^2}, \tag{18.83}$$

H_{max} being the maximum diameter of the subdomains and H_{min} the minimum one.

Remark 18.4 (Approximation of the inverse of A). The inverse of the block matrix A in (18.74) admits the following LDU factorization

$$A^{-1} = \begin{bmatrix} I & -A_{II}^{-1}A_{I\Gamma} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{II}^{-1} & 0 \\ 0 & \Sigma^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{\Gamma I}A_{II}^{-1} & I \end{bmatrix} \tag{18.84}$$

Should we have suitable preconditioners B_{II} of A_{II} and P of Σ, an approximation of A^{-1} would be given by

$$P_A^{-1} = \begin{bmatrix} I & -B_{II}^{-1}A_{I\Gamma} \\ 0 & I \end{bmatrix} \begin{bmatrix} B_{II}^{-1} & 0 \\ 0 & P^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{\Gamma I}B_{II}^{-1} & I \end{bmatrix}. \tag{18.85}$$

An application of P_A^{-1} to a given vector involves B_{II}^{-1} in two matrix-vector multiplies and P^{-1} in only one matrix-vector multiply (see [TW05, Sect. 4.3]). •

Remark 18.5 (Saddle-point systems). In case of a saddle-point (block) matrix like the one in (16.58), an LDU factorization can be obtained as follows

$$K = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} = \begin{bmatrix} I_A & 0 \\ BA^{-1} & I_C \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I_A & A^{-1}B^T \\ 0 & I_C \end{bmatrix} \tag{18.86}$$

where $S = -C - BA^{-1}B^T$ is the Schur complement computed with respect to the second variable (e.g. $\mathbf{P}$ in the case of system (16.58)).
An inverse of K is obtained as

$$K^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & 0 \end{bmatrix} + QS^{-1}Q^T, \quad Q = \begin{bmatrix} -A^{-1}B^T \\ I \end{bmatrix}, \tag{18.87}$$

being I_A and I_C two identity matrices having the size of A and C, respectively. A preconditioner for K can be constructed by replacing in (18.87) A^{-1} and S^{-1} by suitable domain decomposition preconditioners of A and S, respectively.

This observation stands at the ground of the design of the so-called FETI-DP and BDDC preconditioners, see Sects. 18.5.5 and 18.5.6. •

18.4.1 Some numerical results

Consider the Poisson problem (3.13) on the domain $\Omega = (0,1)^2$ whose finite element approximation was given in (4.40).
Let us partition Ω into M disjoint squares Ω_i whose sidelength is H, such that $\cup_{i=1}^{M} \overline{\Omega_i} = \overline{\Omega}$. An example with four subdomains is displayed in Fig. 18.5 (left).

In Table 18.1 we report the numerical values of $K_2(\Sigma)$ for the problem at hand, for several values of the finite element grid-size h; it grows linearly with $1/h$ and with $1/H$, as predicted by the formula (18.83). In Fig. 18.5 (right) we display the *pattern* of the Schur complement matrix Σ in the particular case of $h = 1/8$ and $H = 1/2$. The matrix has a blockwise structure that accounts for the interfaces Γ_1, Γ_2, Γ_3 and Γ_4, plus the contribution arising from the crosspoint Γ_c. Since Σ is a *dense* matrix, when solving the linear system (18.79) the explicit computation of its entries is not convenient. Instead, we can use the following **Algorithm 18.1** to compute the matrix-vector product $\Sigma \mathbf{x}_\Gamma$, for any vector $\mathbf{x}_\Gamma$ (and therefore the residue at every step of an iterative algorithm). We have denoted by R_{Γ_i} the rectangular matrix associated to the restriction operator $R_{\Gamma_i} : \Gamma \to \Gamma_i = \partial\Omega_i \setminus \partial\Omega$, while $\mathbf{x} \leftarrow \mathbf{y}$ indicates the algebraic operation $\mathbf{x} = \mathbf{x} + \mathbf{y}$.

Table 18.1. Condition number of the Schur complement Σ

$K_2(\Sigma)$	$H = 1/2$	$H = 1/4$	$H = 1/8$
$h = 1/8$	9.77	14.83	25.27
$h = 1/16$	21.49	35.25	58.60
$h = 1/32$	44.09	75.10	137.73
$h = 1/64$	91.98	155.19	290.43

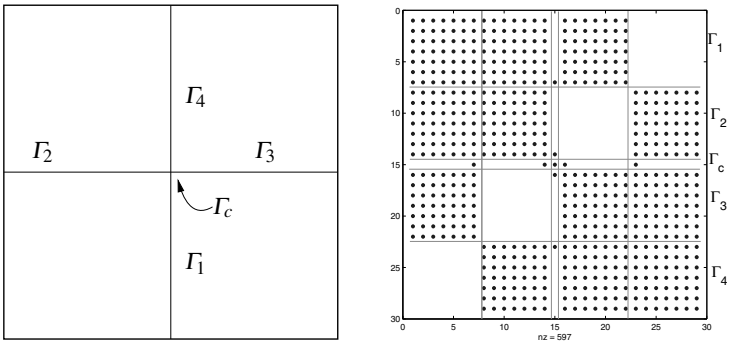

Fig. 18.5. Example of partition of $\Omega = (0,1)^2$ into four squared subdomains (left). Pattern of the Schur complement Σ (right) corresponding to the domain partition displayed on the left

Algorithm 18.1 (Schur complement multiplication by a vector)

Given $\mathbf{x}_\Gamma$, compute $\mathbf{y}_\Gamma = \Sigma \mathbf{x}_\Gamma$ as follows:

a. Set $\mathbf{y}_\Gamma = \mathbf{0}$

b. For $i = 1, \ldots, M$ Do in parallel:

c. $\quad \mathbf{x}_i = R_{\Gamma_i} \mathbf{x}_\Gamma$

d. $\quad \mathbf{z}_i = A_{\Omega_i, \Gamma_i} \mathbf{x}_i$

e. $\quad \mathbf{z}_i \leftarrow A^{-1}_{\Omega_i, \Omega_i} \mathbf{z}_i$

f. $\quad$ sum up in the local vector $\mathbf{y}_{\Gamma_i} \leftarrow A_{\Gamma_i, \Gamma_i} \mathbf{x}_i - A_{\Gamma_i, \Omega_i} \mathbf{z}_i$

g. $\quad$ sum up in the global vector $\mathbf{y}_\Gamma \leftarrow R^T_{\Gamma_i} \mathbf{y}_{\Gamma_i}$

h. EndFor

Since no communication is required among the subdomains, this is a fully parallel algorithm.
Before using for the first time the Schur complement, a start-up phase, described in **Algorithm 18.2**, is requested. Note that this is an *off-line* procedure.

Algorithm 18.2 (Start-up phase for the solution of the Schur complement system)

Given $\mathbf{x}_\Gamma$, compute $\mathbf{y}_\Gamma = \Sigma \mathbf{x}_\Gamma$ as follows:

a. For $i = 1, \ldots, M$ Do in parallel:

b. $\quad$ Compute the entries of A_i

c. $\quad$ Reorder A_i as in (18.76) then extract the submatrices
$A_{\Omega_i, \Omega_i}, A_{\Omega_i, \Gamma_i}, A_{\Gamma_i, \Omega_i}$ and $A^{(i)}_{\Gamma, \Gamma}$

d. $\quad$ Compute the (either LU or Cholesky) factorization of A_{Ω_i, Ω_i}

e. EndFor

18.5 DD preconditioners in case of many subdomains

Before introducing the preconditioners for the Schur complement in the case in which Ω is partitioned in many subdomains we recall the following definition:

Definition 18.1. *A preconditioner P_h of Σ is said to be* scalable *if the condition number of the preconditioned matrix $P_h^{-1}\Sigma$ is independent of the number of subdomains.*

Iterative methods using scalable preconditioners allow henceforth to achieve convergence rates independent of the subdomain number. This is a very desirable property in those cases where a large number of subdomains is used.

Let R_i be a *restriction operator* which, to any vector $\mathbf{v}_h$ of nodal values on the global domain Ω, associates its restriction to the subdomain Ω_i

$$R_i : \mathbf{v}_{h|_\Omega} \rightarrow \mathbf{v}^i_{h|_{\Omega_i \cup \Gamma_i}}.$$

Let moreover

$$R_i^T : \mathbf{v}^i_{h|_{\Omega_i \cup \Gamma_i}} \rightarrow \mathbf{v}_{h|_\Omega}$$

be the *prolongation (or extension-by-zero) operator*. In algebraic form R_i can be represented by a matrix that coincides with the identity matrix in correspondence with the subdomain Ω_i

$$R_i = \left[\begin{array}{ccc|c|ccc} 0 & \dots & 0 & 1 & & & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \ddots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & & & 1 & 0 & \dots & 0 \end{array}\right]. \underbrace{\qquad\qquad}_{\Omega_i}$$

Similarly we can define the restriction and prolongation operators R_{Γ_i} and $R^T_{\Gamma_i}$, respectively, that act on the vector of interface nodal values (as done in (18.81)). In order to find a preconditioner for Σ the strategy consists of combining the contributions of local subdomain preconditioners with that of a global contribution referring to a coarse grid whose elements are the subdomains themselves. Without the latter coarse grid term the preconditioner could not be scalable since it would lack any mechanism for global communication of information across the domain in each iteration step. This idea can be formalized through the following relation that provides the inverse of the preconditioner

$$(P_h)^{-1} = \sum_{i=1}^{M} R^T_{\Gamma_i} P^{-1}_{i,h} R_{\Gamma_i} + R^T_\Gamma P^{-1}_H R_\Gamma \ .$$

We have denoted by H the maximum value of the diameters H_i of the subdomains Ω_i; moreover, $P_{i,h}$ is either the local Schur complement Σ_i, or (more frequently) a suitable preconditioner of Σ_i, while R_Γ and P_H refer to operators that act on the global scale (that of the coarse grid).

Many different choices are possible for the local Schur complement preconditioner $P_{i,h}$; they will give rise to different condition numbers of the preconditioned matrix $P_h^{-1}\Sigma$.

18.5.1 Jacobi preconditioner

Let $\{e_1, \dots, e_m\}$ be the set of edges and $\{v_1, \dots, v_n\}$ that of vertices of a partition of Ω into subdomains (see Fig. 18.6 for an example).

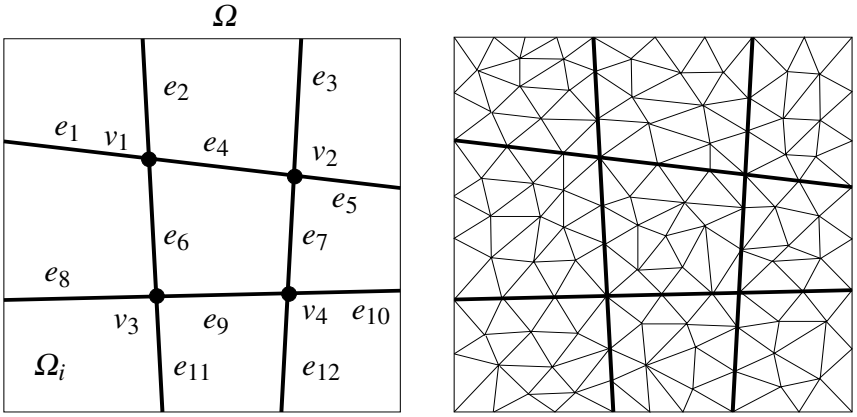

Fig. 18.6. A decomposition into 9 subdomains (left) with a fine triangulation in small triangles and a coarse triangulation in large quadrilaterals (the 9 subdomains) (right)

The Schur complement Σ features the following blockwise representation

$$\Sigma = \left[\begin{array}{c|c} \Sigma_{ee} & \Sigma_{ev} \\ \hline \Sigma_{ev}^T & \Sigma_{vv} \end{array}\right],$$

having set

$$\Sigma_{ee} = \begin{bmatrix} \Sigma_{e_1e_1} & \dots & \Sigma_{e_1e_m} \\ \vdots & \ddots & \vdots \\ \Sigma_{e_me_1} & \dots & \Sigma_{e_me_m} \end{bmatrix}, \quad \Sigma_{ev} = \begin{bmatrix} \Sigma_{e_1v_1} & \dots & \Sigma_{e_1v_n} \\ \vdots & \ddots & \vdots \\ \Sigma_{e_mv_1} & \dots & \Sigma_{e_mv_n} \end{bmatrix}$$

and

$$\Sigma_{vv} = \begin{bmatrix} \Sigma_{v_1v_1} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \Sigma_{v_nv_n} \end{bmatrix}.$$

In 3D there should be a further block row and column due to the presence of faces.

The *Jacobi preconditioner* of the Schur complement Σ is a block diagonal matrix defined by

$$P_h^J = \left[\begin{array}{c|c} \hat{\Sigma}_{ee} & 0 \\ \hline 0 & \Sigma_{vv} \end{array}\right]$$

where $\hat{\Sigma}_{ee}$ is either Σ_{ee} or a suitable approximation of it. This preconditioner does not account for the interaction between the basis functions associated with edges and those

associated with vertices. The matrix $\hat{\Sigma}_{ee}$ is also diagonal

$$\hat{\Sigma}_{ee} = \begin{bmatrix} \hat{\Sigma}_{e_1e_1} & 0 & \dots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \hat{\Sigma}_{e_me_m} \end{bmatrix}.$$

Here $\hat{\Sigma}_{e_ke_k}$ denotes $\Sigma_{e_ke_k}$ or a suitable approximation of it.

The preconditioner P_h^J can also be expressed in terms of restriction and prolongation operators as follows

$$\left(P_h^J\right)^{-1} = \sum_{k=1}^{m} R_{e_k}^T \hat{\Sigma}_{e_ke_k}^{-1} R_{e_k} + R_v^T \Sigma_{vv}^{-1} R_v \,, \tag{18.88}$$

where R_{e_k} and R_v denote edge and vertices restriction operators, respectively.

Regarding the condition number of the preconditioned Schur complement, there exists a constant $C > 0$, indipendent of both h and H, such that

$$K_2\left((P_h^J)^{-1}\Sigma\right) \leq CH^{-2}\left(1 + \log\frac{H}{h}\right)^2.$$

Should the conjugate gradient method be used to solve the preconditioned Schur complement system (18.79) with preconditioner P_h^J, the number of iterations necessary to converge (within a prescribed tolerance) would be proportional to H^{-1}. The presence of H indicates that the Jacobi preconditioner is not scalable.

Moreover, we notice that the presence of the logarithmic term $\log(H/h)$ introduces a relation between the size of the subdomains and the size of the computational grid $\mathcal{T}_h$. This generates a propagation of information among subdomains characterized by a finite (rather than infinite) speed of propagation. Note that the ratio H/h measures the maximum number of elements across any subdomain.

18.5.2 Bramble-Pasciak-Schatz preconditioner

With the aim of accelerating the speed of propagation of information among subdomains we can devise a mechanism of global coupling among subdomains. As already anticipated, the family of subdomains can be regarded as a *coarse* grid, say $\mathcal{T}_H$, of the original domain. For instance, in Fig. 18.6 $\mathcal{T}_H$ is made of 9 (macro) elements and 4 internal nodes. It identifies a stiffness matrix of piecewise bilinear elements, say A_H, of dimension 4×4 which guarantees a global coupling in Ω. We can now introduce a restriction operator that, for simplicity, we indicate $R_H : \Gamma_h \to \Gamma_H$. More precisely, this operator transforms a vector of nodal values on the skeleton Γ_h into a vector of nodal values on the internal vertices of the coarse grid (4 in the case at hand). Its transpose

R_H^T is an extension operator. The matrix P_h^{BPS}, whose inverse is

$$(P_h^{BPS})^{-1} = \sum_{k=1}^{m} R_{e_k}^T \hat{\Sigma}_{e_k e_k}^{-1} R_{e_k} + R_H^T A_H^{-1} R_H \ , \tag{18.89}$$

is named Bramble-Pasciak-Schatz preconditioner. The main difference with Jacobi preconditioner (18.88) is due to the presence of the global (coarse-grid) stiffness matrix A_H instead of the diagonal vertex matrix Σ_{vv}. The following results hold:

$$K_2\left((P_h^{BPS})^{-1}\Sigma\right) \leq C\left(1+\log\frac{H}{h}\right)^2 \text{ in 2D},$$

$$K_2\left((P_h^{BPS})^{-1}\Sigma\right) \leq C\frac{H}{h} \text{ in 3D}.$$

Note that the factor H^{-2} does not show up anymore. The number of iterations of the conjugate gradient method with preconditioner P_h^{BPS} is now proportional to $\log(H/h)$ in 2D and to $(H/h)^{1/2}$ in 3D.

18.5.3 Neumann-Neumann preconditioner

Although the Bramble-Pasciak-Schatz preconditioner has better properties than Jacobi's, yet in 3D the condition number of the preconditioned Schur complement still contains a linear dependence on H/h.

In this respect, a further improvement is achievable using the so-called Neumann-Neumann preconditioner, whose inverse has the following expression

$$(P_h^{NN})^{-1} = \sum_{i=1}^{M} R_{\Gamma_i}^T D_i \Sigma_i^* D_i R_{\Gamma_i}. \tag{18.90}$$

As before, R_{Γ_i} denotes the restriction from the nodal values on the whole skeleton Γ to those on the local interface Γ_i, whereas Σ_i^* is either Σ_i^{-1} (should the local inverse exist) or an approximation of Σ_i^{-1}, e.g. the pseudo-inverse Σ_i^+ of Σ_i. The matrix D_i is a diagonal matrix of positive weights $d_j > 0$, for $j = 1,\ldots,n$, n being the number of nodes on Γ_i. For instance, d_j coincides with the inverse of the number of subdomains that share the $j-th$ node. If we still consider the 4 internal vertices of Fig. 18.6, we will have $d_j = 1/4$, for $j = 1,\ldots,4$.

For the preconditioner (18.90) the following estimate (similar to that of Jacobi preconditioner) holds: there exists a constant $C > 0$, indipendent of both h and H, such that

$$K_2\left((P_h^{NN})^{-1}\Sigma\right) \leq CH^{-2}\left(1+\log\frac{H}{h}\right)^2.$$

The last (logarithmic) factor drops out in case the subdomains partition features no cross points.

The presence of D_i and R_{Γ_i} in (18.90) only entails matrix-matrix multiplications. On the other hand, if $\Sigma_i^* = \Sigma_i^{-1}$, applying Σ_i^{-1} to a given vector can be reconducted to the

use of local inverses. As a matter of fact, let $\mathbf{q}$ be a vector whose components are the nodal values on the local interface Γ_i; then

$$\Sigma_i^{-1}\mathbf{q} = [0,I]A_i^{-1}[0,I]^T\mathbf{q}.$$

In particular, $[0,I]^T\mathbf{q} = [0,\mathbf{q}]^T$, and the matrix-vector product

$$\underbrace{\left[\begin{array}{c|c} \begin{array}{c}\text{internal}\\ \text{nodes}\end{array} & \\ \hline \text{boundary nodes} & \end{array}\right]}_{A_i^{-1}} \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ \hline \mathbf{q} \end{array}\right]$$

corresponds to the solution on Ω_i of the Neumann boundary-value problem

$$\begin{cases} -\triangle w_i = 0 & \text{in } \Omega_i, \\ \dfrac{\partial w_i}{\partial n} = q & \text{on } \Gamma_i. \end{cases} \tag{18.91}$$

Algorithm 18.3 (Neumann-Neumann preconditioner)

Given a vector $\mathbf{r}_\Gamma$, compute $\mathbf{z}_\Gamma = (P_h^{NN})^{-1}\mathbf{r}_\Gamma$ as follows:

a. Set $\mathbf{z}_\Gamma = \mathbf{0}$

b. For $i = 1,\ldots,M$ Do in parallel:

c. restrict the residue on Ω_i: $\mathbf{r}_i = R_{\Gamma_i}\mathbf{r}_\Gamma$

d. compute $\mathbf{z}_i = [0,I]A_i^{-1}[0,\mathbf{r}_i]^T$

e. Sum up the global residue: $\mathbf{z}_\Gamma \leftarrow R_{\Gamma_i}^T\mathbf{z}_i$

f. EndFor

Also in this case a start-up phase is required, consisting in the preparation for the solution of linear systems with local stiffness matrices A_i. Note that in the case of the model problem (3.13), A_i is singular if Ω_i is an internal subdomain, that is if $\partial\Omega_i \setminus \partial\Omega = \emptyset$. One of the following strategies should be adopted:

1. compute a (either LU or Cholesky) factorization of $A_i + \varepsilon I$, for a given $\varepsilon > 0$ sufficiently small;
2. compute a factorization of $A_i + \dfrac{1}{H^2}M_i$, where M_i is the mass matrix whose entries are
$$(M_i)_{k,j} = \int_{\Omega_i} \varphi_k\varphi_j\, d\Omega_i;$$
3. compute the singular-value decomposition of A_i.

Table 18.2. Condition number of the preconditioned matrix $(P_h^{NN})^{-1}\Sigma$

$K_2((P_h^{NN})^{-1}\Sigma)$	$H=1/2$	$H=1/4$	$H=1/8$	$H=1/16$
$h=1/16$	2.55	15.20	47.60	—
$h=1/32$	3.45	20.67	76.46	194.65
$h=1/64$	4.53	26.25	105.38	316.54
$h=1/128$	5.79	31.95	134.02	438.02

The matrix Σ_i^* is defined accordingly. In our numerical results we have adopted the third approach.

The convergence history of the preconditioned conjugate gradient method with preconditioner P_h^{NN} in the case $h=1/32$ is displayed in Fig. 18.7. In Table 18.2 we report the values of the condition number of $(P_h^{NN})^{-1}\Sigma$ for several values of H.

As already pointed out, the Neumann-Neumann preconditioner of the Schur complement matrix is not scalable. A substantial improvement of (18.90) can be achieved by adding a coarse grid correction mechanism, yielding the following new preconditioned Schur complement matrix (see, e.g., [TW05, Sect. 6.2.1])

$$\left(P_h^{BNN}\right)^{-1}\Sigma = P_0 + (I-P_0)((P_h^{NN})^{-1}\Sigma)(I-P_0), \tag{18.92}$$

in which we have used the shorthand notation $P_0 = \bar{R}_0^T\Sigma_0^{-1}\bar{R}_0\Sigma$, $\Sigma_0 = \bar{R}_0\Sigma\bar{R}_0^T$, and $\bar{R}_0$ denotes restriction from Γ onto the coarse level skeleton.
The matrix P_h^{BNN} is called *balanced Neumann-Neumann preconditioner*.

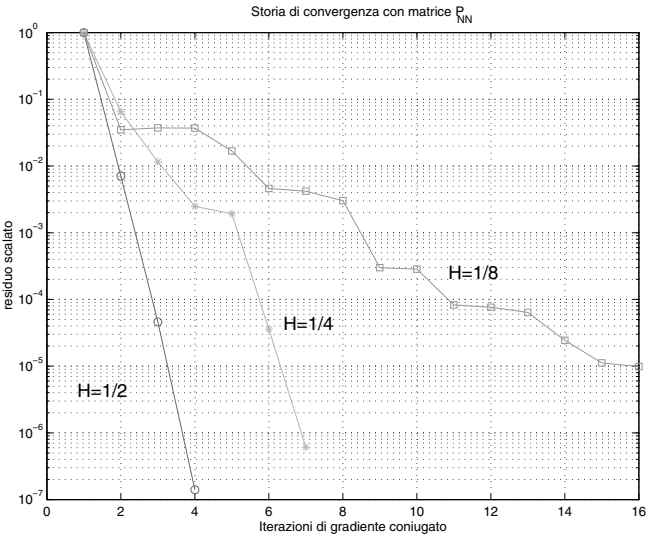

Fig. 18.7. Convergence history for the preconditioned conjugate gradient method with preconditioner P_h^{NN} when $h=1/32$

It can be proven that there exists a constant $C > 0$, independent of h and H, such that

$$K_2\left((P_h^{BNN})^{-1}\Sigma\right) \leq C\left(1+\log\frac{H}{h}\right)^2$$

both in 2D and 3D. The balanced Neumann-Neumann preconditioner therefore guarantees optimal scalability up to a light logarithmic dependence on H and h.
The coarse grid matrix Σ_0 that is a constituent of Σ_H can be built up using the Algorithm 18.4:

Algorithm 18.4 (construction of the coarse matrix for preconditioner P_h^{BNN})

a. Build the restriction operator $\bar{R}_0$ that returns, for every subdomain, the weighted sum of the values at all the nodes at the boundary of that subdomain
For every node the corresponding weight is given by the inverse of the number of subdomains sharing that node

b. Build up the matrix $\Sigma_0 = \bar{R}_0 \Sigma \bar{R}_0^T$

Step a. of this Algorithm is computationally very cheap, whereas step b. requires several (e.g., ℓ) matrix-vector products involving the Schur complement matrix Σ. Since Σ is never built explicitly, this involves the finite element solution of $\ell \times M$ Dirichlet boundary value problems to generate A_H. Observe moreover that the restriction operator introduced at step a. implicitly defines a coarse space whose functions are piecewise constant on every Γ_i. For this reason the balanced Neumann-Neumann preconditioner is especially convenient when either the finite element grid or the subdomain partition (or both) are unstructured, as in Fig. 18.8). An algorithm that implements the BNN preconditioner within a conjugate gradient method to solve the interface problem (18.79) is reported in [TW05, Sect. 6.2.2].

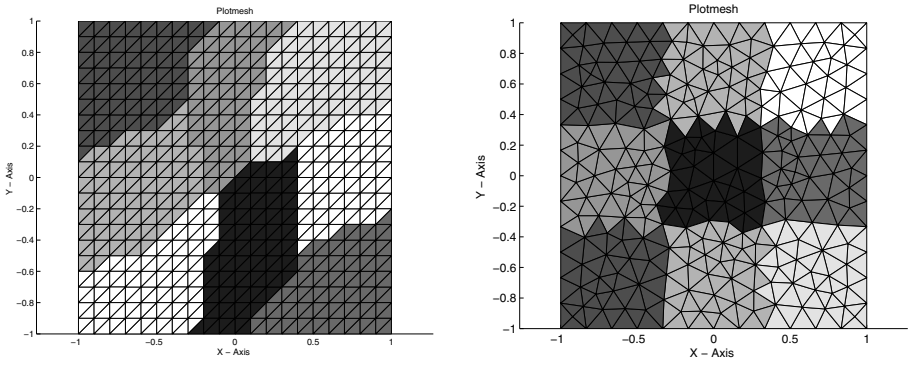

Fig. 18.8. Example of an unstructured subdomain partition in 8 subdomains for a finite element grid which is either structured (left) or unstructured (right)

Table 18.3. Condition number of $(P_h^{BNN})^{-1}\Sigma$ for several values of H

$K_2((P_h^{BNN})^{-1}\Sigma)$	$H=1/2$	$H=1/4$	$H=1/8$	$H=1/16$
$h=1/16$	1.67	1.48	1.27	—
$h=1/32$	2.17	2.03	1.47	1.29
$h=1/64$	2.78	2.76	2.08	1.55
$h=1/128$	3.51	3.67	2.81	2.07

By a comparison of the results obtained using the Neumann-Neumann preconditioner (with and without balancing), the following conclusions can be drawn:

- although featuring a better condition number than A, Σ is still ill-conditioned. The use of a suitable preconditioner is therefore mandatory;
- the Neumann-Neumann preconditioner can be satisfactorily used for partitions featuring a moderate number of subdomains;
- the balancing Neumann-Neumann preconditioner is almost optimally scalable and therefore recommandable for partitions with a large number of subdomains.

18.5.4 FETI (Finite Element Tearing & Interconnecting) methods

In this section we will denote by $H_i = diam(\Omega_i)$, $W_i = W^h(\partial\Omega_i)$ (the space of traces of finite element functions on the boundaries $\partial\Omega_i$), and by $W = \prod_{i=1}^{M} W_i$ the product space of such trace spaces.

At a later stage we will need two further finite element trace spaces, $\widehat{W} \subset W$ a subspace of *continuous traces* across the skeleton Γ, and $\widetilde{W}$, a possible intermediate space $\widehat{W} \subset \widetilde{W} \subset W$ that will fulfill a smaller number of continuity constraints.

We will consider the variable coefficient elliptic problem

$$\begin{cases} -div(\rho\nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{18.93}$$

where ρ is piecewise constant, $\rho = \rho_i \in \mathbb{R}^+$ in Ω_i.
Finally, we will denote by Ω_{ih} the nodes in Ω_i, $\partial\Omega_{ih}$ the nodes on $\partial\Omega_i$, $\partial\Omega_h$ the nodes on $\partial\Omega$, and Γ_h the nodes on Γ. See Fig. 18.9.

Let us introduce the following scaling counting functions: $\forall x \in \Gamma_h \cup \partial\Omega_h$

$$\delta_i(x) = \begin{cases} 1 & x \in \partial\Omega_{ih} \cap (\partial\Omega_h \setminus \Gamma_h), \\ \displaystyle\sum_{j \in N_x} \rho_j^{\gamma}(x)/\rho_i^{\gamma}(x) & x \in \partial\Omega_{ih} \cap \Gamma_h, \\ 0 & \text{elsewhere} \end{cases} \tag{18.94}$$

where $\gamma \in [1/2, +\infty)$ and N_x is the set of indices of the subregions having x on their boundary. Then we set

$$\delta_i^{\dagger}(x) \quad (= \text{ pseudo inverses }) = \begin{cases} \delta_i^{-1}(x) & \text{if } \delta_i(x) \neq 0, \\ 0 & \text{if } \delta_i(x) = 0. \end{cases} \tag{18.95}$$

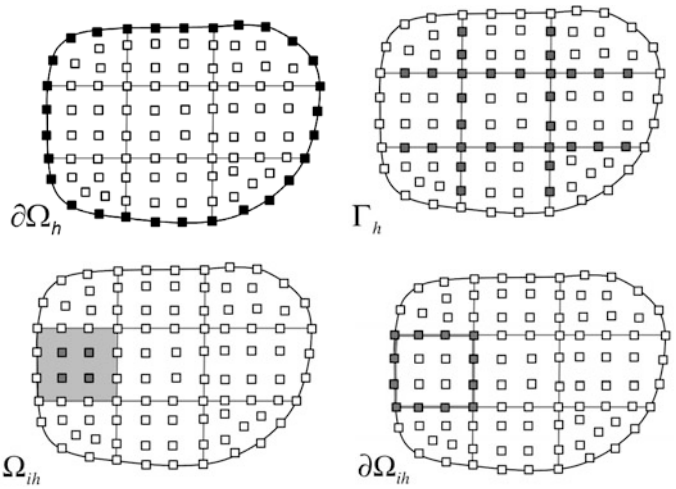

Fig. 18.9. Finite element sets of nodes $\partial\Omega_h$, Γ_h, Ω_{ih}, and $\partial\Omega_{ih}$

Based on the finite element approximation of (18.93), let us consider the local Schur complements (18.80), which are positive semi-definite matrices. In this section we will indicate the interface nodal values on $\partial\Omega_i$ as $\mathbf{u}_i$, and we set $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_M)$, the local load vectors on $\partial\Omega_i$ as $\boldsymbol{\chi}_i$ and we set $\boldsymbol{\chi}_\Delta = (\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_M)$. Finally, we set

$$\Sigma_\Delta = diag(\Sigma_1, \ldots, \Sigma_M) = \begin{bmatrix} \Sigma_1 & 0 & \cdots & 0 \\ \vdots & \Sigma_2 & & \vdots \\ & & \ddots & \\ 0 & 0 & \cdots & \Sigma_M \end{bmatrix},$$

a block diagonal matrix.
The original FEM problem, when reduced to the interface Γ, reads

$$\begin{cases} \text{Find } \mathbf{u} \in W \text{ such that} & J(\mathbf{u}) = \dfrac{1}{2}\langle \Sigma_\Delta \mathbf{u}, \mathbf{u}\rangle - \langle \boldsymbol{\chi}_\Delta, \mathbf{u}\rangle \to min, \\ & B_\Gamma \mathbf{u} = \mathbf{0}. \end{cases} \tag{18.96}$$

B_Γ is not unique, so that we should impose continuity when $\mathbf{u}$ belongs to more than one subdomain; B_Γ is made of $\{0, -1, 1\}$, since it enforces continuity constraints at interfaces' nodes. Here, we are using the same notation W to denote the finite element space trace and that of their nodal values at points of Γ_h.
In 2D, there is a little choice on how to write the constraint of continuity at a point sitting on an edge, there are many options for a vertex point. For the edge node we only need to choose the sign, whereas for a vertex node, e.g. one common to 4 subdomains, a minimum set of three constraints can be chosen in many different ways to assure continuity at the node in question. See, e.g., Fig. 18.10.

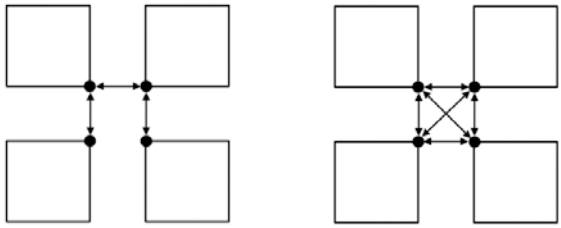

Fig. 18.10. Continuity constraints enforced by 3 (non-redundant) conditions on the left, by 6 (redundant) conditions on the right

Problem (18.96) admits a unique solution iff $Ker\{\Sigma_\Delta\} \cap Ker\{B_\Gamma\} = 0$, that is if Σ_Δ is invertible on $Ker(B_\Gamma)$.

We can reformulate (18.96) using Lagrange multipliers:

$$\begin{cases} \text{Find } (\mathbf{u},\boldsymbol{\lambda}) \in W \times U \text{ such that} & \Sigma_\Delta \mathbf{u} + B_\Gamma^T \boldsymbol{\lambda} = \boldsymbol{\chi}_\Delta, \\ & B_\Gamma \mathbf{u} = \mathbf{0}. \end{cases} \tag{18.97}$$

Because of the inf-sup (LBB) condition (see Chap. 16), the component $\boldsymbol{\lambda}$ of the solution to (18.97) is unique up to an additive vector of $Ker(B_\Gamma^T)$, so we choose $U = range(B_\Gamma)$. Let $R = diag(R^{(1)},\ldots,R^{(M)})$ be made of null-space elements of Σ_Δ. (E.g. $R^{(i)}$ corresponds to the rigid body motions of Ω_i, in case of linear elasticity operator.)
R is a full column rank matrix. The solution of the first equation of (18.97) exists iff $\boldsymbol{\chi}_\Delta - B_\Gamma^T \boldsymbol{\lambda} \in range(\Sigma_\Delta)$, a limitation that will be resolved by introducing a suitable projection operator P. Then,

$$\mathbf{u} = \Sigma_\Delta^\dagger (\boldsymbol{\chi}_\Delta - B_\Gamma^T \boldsymbol{\lambda}) + R\boldsymbol{\alpha} \quad \text{ if } \boldsymbol{\chi}_\Delta - B_\Gamma^T \boldsymbol{\lambda} \perp Ker(\Sigma_\Delta),$$

where $\boldsymbol{\alpha}$ is an arbitrary vector and $\Sigma_\Delta^\dagger$ is a pseudoinverse of Σ_Δ. (Even though there are several pseudo-inverses of a given matrix, the following algorithm will be invariant to the specific choice.) It is convenient to choose a symmetric $\Sigma_\Delta^\dagger$, e.g. that of Moore-Penrose, see [QSS07].
Substituting $\mathbf{u}$ into the second equation of (18.97) yields

$$B_\Gamma \Sigma_\Delta^\dagger B_\Gamma^T \boldsymbol{\lambda} = B_\Gamma \Sigma_\Delta^\dagger \boldsymbol{\chi}_\Delta + B_\Gamma R\boldsymbol{\alpha}. \tag{18.98}$$

Let us set $F = B_\Gamma \Sigma_\Delta^\dagger B_\Gamma^T$ and $\mathbf{d} = B_\Gamma \Sigma_\Delta^\dagger \boldsymbol{\chi}_\Delta$. Then choose P^T to be a suitable projection matrix, e.g. $P^T = I - G(G^T G)^{-1} G^T$, with $G = B_\Gamma R$. Then

$$\begin{cases} P^T F \boldsymbol{\lambda} = P^T \mathbf{d}, \\ G^T \boldsymbol{\lambda} = \mathbf{e}\ (= R^T \boldsymbol{\chi}_\Delta). \end{cases} \tag{18.99}$$

More in general, one can introduce a s.p.d. matrix Q, and set

$$P^T = I - G(G^T Q G)^{-1} G^T Q.$$

The operator P^T is the projection from U onto the space of Lagrange multipliers that are Q-orthogonal to $range(G)$, while $P = I - QG(G^TQG)^{-1}G^T$ is a projection from U onto $Ker(G^T)$ (it is indeed the orthogonal projection with respect to the Q^{-1}-inner product $(\boldsymbol{\lambda}, \boldsymbol{\mu})_{Q^{-1}} = (\boldsymbol{\lambda}, Q^{-1}\boldsymbol{\mu})$).
Upon multiplication of (18.98) by $H = (G^TQG)^{-1}G^TQ$ we find

$$\boldsymbol{\alpha} = H(\mathbf{d} - F\boldsymbol{\lambda}),$$

which fully determines the primal variables in terms of $\boldsymbol{\lambda}$.
If the differential operator has constant coefficients, choosing $Q = I$ suffices. In case of jumps in the coefficients, Q is typically chosen as a scaling diagonal matrix and can be regarded as a scaling from the left of matrix B_Γ by $Q^{1/2}$.

The original one-level FETI method is a CG method in the space V applied to

$$P^TF\boldsymbol{\lambda} = P^T\mathbf{d}, \quad \boldsymbol{\lambda} \in \boldsymbol{\lambda}_0 + V \tag{18.100}$$

with an initial $\boldsymbol{\lambda}_0$ such that $G^T\boldsymbol{\lambda}_0 = \mathbf{e}$. Here

$$V = \{\boldsymbol{\lambda} \in U : \langle \boldsymbol{\lambda}, B\mathbf{z}\rangle = 0, \mathbf{z} = Ker(\Sigma_\Delta)\}$$

is the so-called space of admissible increments, $Ker(G^T) = range(P)$ and

$$V' = \{\boldsymbol{\mu} \in U : \langle \boldsymbol{\mu}, B\mathbf{z}\rangle_Q = 0, \mathbf{z} \in Ker(\Sigma_\Delta)\} = range(P^T).$$

The above simplest version of FETI with no preconditioner (or only a diagonal preconditioner) in the subdomain is scalable with the number of subdomains, but the condition number grows polynomially with the number of elements per subdomain.
The original, most basic FETI preconditioner is

$$P_h^{-1} = B_\Gamma \Sigma_\Delta B_\Gamma^T = \sum_{i=1}^{M} B^{(i)} \Sigma_i {B^{(i)}}^T. \tag{18.101}$$

It is called a Dirichlet preconditioner since its application to a given vector involves the solution of M independent Dirichlet problems, one in every subdomain. The coarse space in FETI consists of the nullspace on each substructure.

To keep the search directions of the resulting preconditioned CG method in the space V, the application of P_h^{-1} is followed by an application of the projection P. Thus, the so-called Dirichlet variant of the FETI method is the CG algorithm applied to the modified equation

$$PP_h^{-1}P^TF\boldsymbol{\lambda} = PP_h^{-1}P^T\mathbf{d}, \quad \boldsymbol{\lambda} \in \boldsymbol{\lambda}_0 + V. \tag{18.102}$$

Since, for $\boldsymbol{\lambda} \in V$, $PP_h^{-1}P^TF\boldsymbol{\lambda} = PP_h^{-1}P^TP^TFP\boldsymbol{\lambda}$, the matrix on the left of (18.102) can be regarded as the product of two symmetric matrices. In case B_Γ has full row rank, i.e. the constraints are linearly independent and there are no redundant Lagrange multipliers, a better preconditioner can be defined as follows

$$\widehat{P}_h^{-1} = (B_\Gamma D^{-1} B_\Gamma^T)^{-1} B_\Gamma D^{-1} \Sigma_\Delta D^{-1} B_\Gamma^T (B_\Gamma D^{-1} B_\Gamma^T)^{-1} \tag{18.103}$$

where D is a block diagonal matrix $D = diag(D^{(1)}, \ldots, D^{(M)})$ and each block $D^{(i)}$ is a diagonal matrix whose elements are $\delta_i^{\dagger}(x)$ (see (18.95)) corresponding to the point x of $\partial\Omega_{i,h}$.
Since $B_\Gamma D^{-1} B_\Gamma^T$ is block-diagonal, its inverse can be easily computed by inverting small blocks whose size is n_x, the number of Lagrange multipliers used to enforce continuity at point x.
The matrix D, that operates on elements of the product space W, can be regarded as a scaling from the right of B_Γ by $D^{-1/2}$. With this choice

$$K_2(P\widehat{P}_h^{-1}P^T F) \leq C(1 + log(H/h))^2, \tag{18.104}$$

where $K_2(\cdot)$ is the spectral condition number and C is a constant independent of h, H, γ and the values of the ρ_i.

18.5.5 FETI-DP (Dual Primal FETI) methods

The FETI-DP method is a domain decomposition method introduced in [FLT$^+$01] that enforces equality of the solution at subdomains interfaces by Lagrange multipliers except at subdomains corners, which remain primal variables. The first mathematical analysis of the method was provided by Mandel and Tezaur [MT01]. The method was further improved by enforcing the equality of averages across the edges or faces on subdomain interfaces [FLP00], [KWD02]. This is important for parallel scalability.

Let us consider a 2D case for simplicity. As anticipated at the beginning of Sect. 18.5.4, this idea is implemented by introducing an additional space $\widetilde{W}$ such that $\widehat{W} \subset \widetilde{W} \subset W$ for which we have continuity of the primal variables at subdomain vertices, and also common values of the averages over all edges of the interface. However, for simplicity we will confine ourselves to the case of primal variables associated to subdomain vertices only. This space can be written as the sum of two subspaces

$$\widetilde{W} = \widehat{W}_{\Pi} \oplus \widetilde{W}_{\Delta} \tag{18.105}$$

where $\widehat{W}_{\Pi} \subset \widehat{W}$ is the space of continuous interface functions that vanish at all nodal points of Γ_h except at the subdomain vertices. $\widehat{W}_{\Pi}$ is given in terms of the vertex variables and the averages of the values over the individual edges of the set of interface nodes Γ_h. $\widetilde{W}_{\Delta}$ is the direct sum of local subspaces $\widetilde{W}_{\Delta,i}$:

$$\widetilde{W}_{\Delta} = \prod_{i=1}^{M} \widetilde{W}_{\Delta,i} \tag{18.106}$$

where $\widetilde{W}_{\Delta,i} \subset W_i$ consists of local functions on $\partial\Omega_i$ that vanish at the vertces of Ω_i and have zero average on each individual edge.

According to this space splitting, the continuous degrees of freedom associated with the subdomain vertices and with the subspace $\widehat{W}_{\Pi}$ are called *primal* (Π), while those (that are potentially discontinuous across Γ) that are associated with the subspaces $\widetilde{W}_{\Delta,i}$ and with the interior of the subdomain edges are called *dual* (Δ).

The subspace $\widehat{W}_{\Pi}$, together with the interior subspace, defines the subsystem which is fully assembled, factored, and stored in each iteration step.

At this stage, all unknowns of the first subspace as well as the interior variables are eliminated to obtain a new Schur complement $\widetilde{\Sigma}_{\Delta}$. More precisely, we proceed as follows.

Let $\widetilde{A}$ denote the stiffness matrix obtained by restricting $diag(A_1,\ldots,A_M)$ (see (18.76)) from $\prod_{i=1}^{M} W^h(\Omega_i)$ to $\widetilde{W}^h(\Omega)$ (these spaces now refer to subdomains, not to their boundaries). Then $\widetilde{A}$ is no longer block diagonal because of the coupling that now exists between subdomains sharing a common vertex. According to the previous space decomposition, $\widetilde{A}$ can be split as follows

$$\widetilde{A} = \begin{bmatrix} A_{II} & A_{I\Pi} & A_{I\Delta} \\ A_{I\Pi}^T & A_{\Pi\Pi} & A_{\Pi\Delta} \\ A_{I\Delta}^T & A_{\Pi\Delta}^T & A_{\Delta\Delta} \end{bmatrix}.$$

Here the subscript I refers to the internal degrees of freedom of the subdomains, Π to those associated to the subdomains vertices, and Δ to those of the interior of the subdomains edges, see Fig. 18.11, right. The matrices A_{II} and $A_{\Delta\Delta}$ are block diagonal (one block per subdomain). Any non-zero entry of $A_{I\Delta}$ represents a coupling between degrees of freedom associated with the same subdomain. Upon eliminating the variables of the I and Π sets, a Schur complement associated with the variables of the Δ sets (interior and edges) is obtained as follows

$$\widetilde{\Sigma} = A_{\Delta\Delta} - \begin{bmatrix} A_{I\Delta}^T A_{\Pi\Delta}^T \end{bmatrix} \begin{bmatrix} A_{II} & A_{I\Pi} \\ A_{I\Pi}^T & A_{\Pi\Pi} \end{bmatrix}^{-1} \begin{bmatrix} A_{I\Delta} \\ A_{\Pi\Delta} \end{bmatrix}. \tag{18.107}$$

Correspondingly we obtain a reduced right hand side $\tilde{\chi}_{\Delta}$. By indicating with $\mathbf{u}_{\Delta} \in \widetilde{W}_{\Delta}$ the vector of degrees of freedom associated with the edges, similarly to what done in (18.96) for FETI, the finite element problem can be reformulated as a minimization

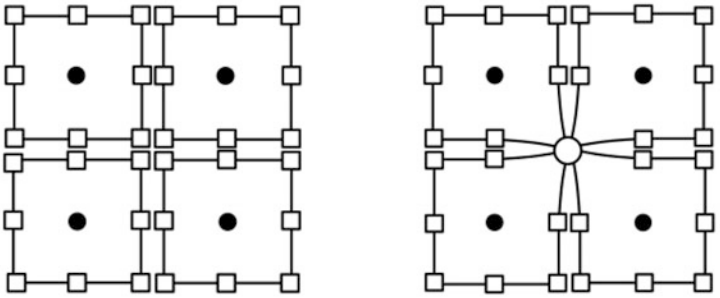

Fig. 18.11. Degrees of freedom of the space W for one-level FETI (left) and those of the space $\widetilde{W}$ for one-level FETI-DP (right) in the case of primal vertices only

problem with constraints given by the requirement of continuity across all of Γ

$$\begin{cases} \text{Find } \mathbf{u}_\Delta \in \widetilde{W}_\Delta : & J(\mathbf{u}_\Delta) &= \frac{1}{2}\langle \widetilde{\Sigma}\mathbf{u}_\Delta, \mathbf{u}_\Delta\rangle - \langle \tilde{\chi}_\Delta, \mathbf{u}_\Delta\rangle \to min, \\ & B_\Delta \mathbf{u}_\Delta &= \mathbf{0}. \end{cases} \tag{18.108}$$

The matrix B_Δ is made of $\{0,-1,1\}$ as it was for B_Γ. Note however that this time the constraints associated with the vertex nodes are dropped since they are assigned to the primal set. Note also that since all the constraints refer to edge points, no distinction needs to be made between redundant and non-redundant constraints and Lagrange multipliers.

A saddle point formulation of (18.108), similar to (18.97), can be obtained by introducing a set of Lagrange multipliers $\boldsymbol{\lambda} \in V = range(B_\Delta)$. Indeed, since $\widetilde{A}$ is s.p.d., so is $\widetilde{\Sigma}$: by eliminating the subvectors $\mathbf{u}_\Delta$ we obtain the reduced system

$$F_\Delta \boldsymbol{\lambda} = \mathbf{d}_\Delta, \tag{18.109}$$

where $F_\Delta = B_\Delta \widetilde{\Sigma}^{-1} B_\Delta^T$ and $\mathbf{d}_\Delta = B_\Delta \widetilde{\Sigma}^{-1} \tilde{\chi}_\Delta$.

Note that once $\boldsymbol{\lambda}$ is found, $\mathbf{u}_\Delta = \widetilde{\Sigma}^{-1}(\tilde{\chi}_\Delta - B_\Delta^T \boldsymbol{\lambda}) \in \widetilde{W}_\Delta$, while the interior variables $\mathbf{u}_I$ and the vertex variables $\mathbf{u}_\Pi$ are obtained by back-solving the system associated with $\widetilde{A}$.

A preconditioner for F is introduced as done in (18.103) for FETI (in case of non-redundant Lagrange multipliers)

$$P_\Delta^{-1} = (B_\Delta D_\Delta^{-1} B_\Delta^T)^{-1} B_\Delta D_\Delta^{-1} S_{\Delta\Delta} D_\Delta^{-1} B_\Delta^T (B_\Delta D_\Delta^{-1} B_\Delta^T)^{-1}. \tag{18.110}$$

Here D_Δ is a block diagonal scaling matrix with blocks $D_\Delta^{(i)}$: each of their diagonal elements corresponds to a Lagrange multiplier that enforces continuity between the nodal values of some $w_i \in W_i$ and $w_j \in W_j$ at some point $x \in \Gamma_h$ and it is given by $\delta_j^\dagger(x)$. Moreover, $\Sigma_{\Delta\Delta} = diag(\Sigma_{1,\Delta\Delta}, \ldots, \Sigma_{M,\Delta\Delta})$ with $\Sigma_{i,\Delta\Delta}$ being the restriction of the local Schur complement Σ_i to $\widetilde{W}_{\Delta,i} \subset W_i$.

When using the conjugate gradient method for the preconditioned system

$$P_\Delta^{-1} F_\Delta \boldsymbol{\lambda} = P_\Delta^{-1} \mathbf{d}_\Delta,$$

in contrast with one level FETI methods we can use an arbitrary initial guess $\boldsymbol{\lambda}^0$.

For an efficient implementation of this algorithm see [TW05, Sect. 6.4.1]. Also in this case we have a condition number that scales polylogarithmically, that is

$$K_2(P_\Delta^{-1} F_\Delta) \leq C(1 + log(H/h))^2,$$

where C is independent of h, H, γ and the values of the ρ_i. For a comprehensive presentation and analysis, see [KWD02] and [TW05].

For a conclusive comparative remark between FETI and FETI-DP methods, by following [TW05] we can note that FETI-DP algorithms do not require the characterization of the kernels of local Neumann problems (as required by one-level methods), because the enforcement of the additional constraints in each iteration always makes the local problems nonsingular and at the same time provides an underlying coarse global prob-

lem. FETI-DP methods do not require the introduction of a scaling matrix Q, which enters in the construction of a coarse solver for one-level FETI algorithms.
Finally, it is worth noticing that one-level FETI methods are projected conjugate gradient algorithms that cannot start from an arbitrary initial guess. In contrast, FETI-DP methods are standard preconditioned conjugate algorithms and can therefore employ an arbitrary initial guess $\boldsymbol{\lambda}^0$.

18.5.6 BDDC (Balancing Domain Decomposition with Constraints) methods

This method was introduced by Dohrmann [Doh03] as a simpler primal alternative to the FETI-DP domain decomposition method. The name BDDC was coined by Mandel and Dohrmann because it can be understood as further development of the balancing domain decomposition method [Man93] with the coarse, global component of a BDDC algorithm expressed interms of a set of primal constraints.

In contrast to the original Neumann-Neumann and one-levet FETI methods, FETI-DP and BDDC algorithms do not require the solution of any singular linear systems of equations (those associated with a pure Neumann problem). In fact, any given choice of the primal set of variables determines a FETI-DP method and an associated BDDC method. This pair defines a duality, and features the same spectrum of eigenvalues (up to the eigenvalues 0 and 1) (see [LW06]). The choice of the primal constraints is of course a crucial question in order to obtain an efficient FETI-DP or BDDC algorithm.

BDDC is used as a preconditioner for the conjugate gradient method. A specific version of BDDC is characterized by the choice of coarse degrees of freedom, which can be values at the corners of the subdomains, or averages over the edges of the interface between the subdomains. One application of the BDDC preconditioner then combines the solution of local problems on each subdomain with the solution of a global coarse problem with the coarse degrees of freedom as the unknowns. The local problems on different subdomains are completely independent of each other, so the method is suitable for parallel computing.
A BDDC preconditioner reads

$$P_{BDDC}^{-1} = \widetilde{R}_{D\Gamma}^T \widetilde{\Sigma}^{-1} \widetilde{R}_{D\Gamma},$$

where $\widetilde{R}_\Gamma : \widehat{W} \to \widetilde{W}$ is a restriction matrix, $\widetilde{R}_{D\Gamma}$ is a scaled variant of $\widetilde{R}_\Gamma$ with scale factor $\delta_i^\dagger$ (featuring the same sparsity pattern of $\widetilde{R}_\Gamma$). This scaling is chosen in such a way that $\widetilde{R}_\Gamma \widetilde{R}_{D\Gamma}^T$ is a projection (then it coincides with its square).

Theoretical analysis of BDDC preconditioner (and its spectral analogy with FETI-DP preconditioner) was first provided in [MDT05] and later in [LW06] and [BS07].

18.6 Schwarz iterative methods

Schwarz method, in its original form described in Sect. 18.1.1, was proposed by H. Schwarz [Sch69] as an iterative scheme to prove existence of solutions to elliptic equations set in domains whose shape inhibits a direct application of Fourier series.

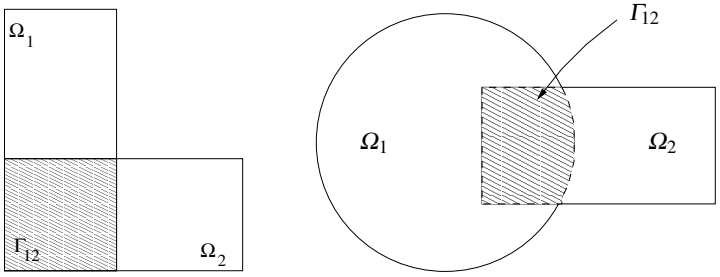

Fig. 18.12. Two examples for which the Schwarz method in its classical form applies

Two elementary examples are displayed in Fig. 18.12. This method is still used in some quarters as solution method for elliptic equations in arbitrarily shaped domains. However, nowadays it is mostly used in a somehow different version, that of DD preconditioner of conjugate gradient (or, more generally, Krylov) iterations for the solution of algebraic systems arising from finite element (or other kind of) discretizations of boundary-value problems.

As seen in Sect. 18.1.1, the distinctive feature of Schwarz method is that it is based on an overlapping subdivision of the original domain. Let us still denote $\{\Omega_m\}$ these subdomains.

To start with, in the following subsection we will show how the Schwarz method can be formulated as an iterative algorithm to solve the algebraic system associated with the finite element discretization of problem (18.1).

18.6.1 Algebraic form of Schwarz method for finite element discretizations

Consider as usual a finite element triangulation $\mathcal{T}_h$ of the domain Ω. Then assume that Ω is decomposed in two overlapping subdomains, Ω_1 and Ω_2, as shown in Fig. 18.1 (left).

Denote with N_h the total number of nodes of the triangulation that are internal to Ω (i.e., they don't sit on its boundary), and with N_1 and N_2, respectively, those internal to Ω_1 and Ω_2, as done in Sect. 18.3. Note that $N_h \leq N_1 + N_2$ and that equality holds only if the overlap reduces to a single layer of elements. Indeed, if we denote with $I = \{1, \ldots, N_h\}$ the set of indices of the nodes of Ω, and with I_1 and I_2 those associated with the internal nodes of Ω_1 and Ω_2, respectively, one has $I = I_1 \cup I_2$, while $I_1 \cap I_2 \neq \emptyset$ unless the overlap consists of a single layer of elements.

Let us order the nodes in such a way that the first block corresponds to those in $\Omega_1 \setminus \Omega_2$, the second to those in $\Omega_1 \cap \Omega_2$, and the third to those in $\Omega_2 \setminus \Omega_1$. The stiffness matrix A of the finite element discretization contains two submatrices, A_1 and A_2, corresponding to the local stiffness matrices in Ω_1 e Ω_2, respectively (see Fig. 18.13). They are related to A as follows

$$A_1 = R_1 A R_1^T \in \mathbb{R}^{N_1 \times N_1} \quad \text{and} \quad A_2 = R_2 A R_2^T \in \mathbb{R}^{N_2 \times N_2}, \tag{18.111}$$

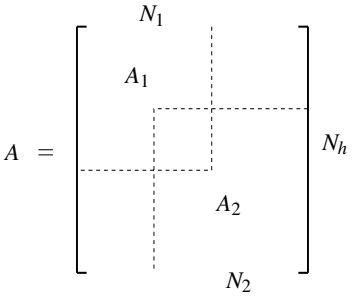

Fig. 18.13. The submatrices A_1 and A_2 of the stiffness matrix A

being R_i and R_i^T, for $i = 1,2$, the restriction and prolongation operators, respectively. The matrix representation of the latter is

$$R_1^T = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \\ & \mathbf{0} & \end{bmatrix} \in \mathbb{R}^{N_h \times N_1}, \qquad R_2^T = \begin{bmatrix} & \mathbf{0} & \\ 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{N_h \times N_2}. \quad (18.112)$$

If $\mathbf{v}$ is a vector of $\mathbb{R}^{N_h}$, then $R_1\mathbf{v}$ is a vector of $\mathbb{R}^{N_1}$ whose components coincide with the first N_1 components of $\mathbf{v}$. Should $\mathbf{v}$ instead be a vector of $\mathbb{R}^{N_1}$, then $R_1^T\mathbf{v}$ would be a vector of dimension N_h whose last $N_h - N_1$ components are all zero.

By using these definitions, an iteration of the multiplicative Schwarz method applied to system $A\mathbf{u} = \mathbf{f}$ can be expressed as follows:

$$\mathbf{u}^{(k+1/2)} = \mathbf{u}^{(k)} + R_1^T A_1^{-1} R_1(\mathbf{f} - A\mathbf{u}^{(k)}), \quad (18.113)$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k+1/2)} + R_2^T A_2^{-1} R_2(\mathbf{f} - A\mathbf{u}^{(k+1/2)}). \quad (18.114)$$

Equivalently, by setting

$$P_i = R_i^T A_i^{-1} R_i A \ , \quad i = 1,2, \quad (18.115)$$

we have

$$\mathbf{u}^{(k+1/2)} = (I - P_1)\mathbf{u}^{(k)} + P_1\mathbf{u},$$

$$\mathbf{u}^{(k+1)} = (I - P_2)\mathbf{u}^{(k+1/2)} + P_2\mathbf{u} = (I - P_2)(I - P_1)\mathbf{u}^{(k)} + (P_1 + P_2 - P_2P_1)\mathbf{u}.$$

Similarly, an iteration of the additive Schwarz method reads

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + (R_1^T A_1^{-1} R_1 + R_2^T A_2^{-1} R_2)(\mathbf{f} - A\mathbf{u}^{(k)}), \quad (18.116)$$

that is

$$\mathbf{u}^{(k+1)} = (I - P_1 - P_2)\mathbf{u}^{(k)} + (P_1 + P_2)\mathbf{u}. \quad (18.117)$$

Introducing the matrices

$$Q_i = R_i^T A_i^{-1} R_i = P_i A^{-1},\ i = 1,2,$$

from (18.113) and (18.114) we derive the following recursive formula for the multiplicative Schwarz method

$$\begin{aligned}\mathbf{u}^{(k+1)} &= \mathbf{u}^{(k)} + Q_1(\mathbf{f} - A\mathbf{u}^{(k)}) + Q_2[\mathbf{f} - A(\mathbf{u}^{(k)} + Q_1(\mathbf{f} - A\mathbf{u}^{(k)}))] \\ &= \mathbf{u}^{(k)} + (Q_1 + Q_2 - Q_2 A Q_1)(\mathbf{f} - A\mathbf{u}^{(k)}),\end{aligned}$$

whereas for the additive Schwarz method we obtain from (18.116) that

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + (Q_1 + Q_2)(\mathbf{f} - A\mathbf{u}^{(k)}). \tag{18.118}$$

This last formula can easily be extended to the case of a decomposition of Ω into $M \geq 2$ overlapping subdomains $\{\Omega_i\}$ (see Fig. 18.14 for an example). In this case we have

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \Big(\sum_{i=1}^{M} Q_i\Big)(\mathbf{f} - A\mathbf{u}^{(k)}). \tag{18.119}$$

18.6.2 Schwarz preconditioners

Denoting with

$$P_{as} = \Big(\sum_{i=1}^{M} Q_i\Big)^{-1}, \tag{18.120}$$

from (18.119) it follows that an iteration of the additive Schwarz method corresponds to an iteration of the preconditioned Richardson method applied to the solution of the linear system $A\mathbf{u} = \mathbf{f}$ using P_{as} as preconditioner. For this reason the matrix P_{as} is named *additive Schwarz preconditioner*.

In case of disjoint subdomains (no overlap), P_{as} coincides with the *block Jacobi preconditioner*

$$P_J = \begin{bmatrix} A_1 & & 0 \\ & \ddots & \\ 0 & & A_M \end{bmatrix}, \quad P_J^{-1} = \begin{bmatrix} A_1^{-1} & & 0 \\ & \ddots & \\ 0 & & A_M^{-1} \end{bmatrix} \tag{18.121}$$

in which we have removed the off-diagonal blocks of A.

Equivalently, one iteration of the additive Schwarz method corresponds to an iteration by the Richardson method on the preconditioned linear system $Q_a\mathbf{u} = \mathbf{g}_a$, with $\mathbf{g}_a = P_{as}^{-1}\mathbf{f}$, and the preconditioned matrix Q_a is

$$Q_a = P_{as}^{-1} A = \sum_{i=1}^{M} P_i.$$

By proceeding similarly, using the multiplicative Schwarz method would yield the following preconditioned matrix

$$Q_M = P_{ms}^{-1}A = I-(I-P_M)\dots(I-P_1).$$

Lemma 18.3. *Matrices P_i defined in (18.115) are symmetric and non-negative w.r.t the following scalar product induced by A*

$$(\mathbf{w},\mathbf{v})_A = (A\mathbf{w},\mathbf{v}) \qquad \forall \mathbf{w},\mathbf{v}\in\mathbb{R}^{N_h}.$$

Proof. For $i=1,2$, we have

$$\begin{aligned}(P_i\mathbf{w},\mathbf{v})_A &= (AP_i\mathbf{w},\mathbf{v}) = (R_i^TA_i^{-1}R_iA\mathbf{w},A\mathbf{v}) = (A\mathbf{w},R_i^TA_i^{-1}R_iA\mathbf{v})\\ &= (\mathbf{w},P_i\mathbf{v})_A \quad \forall \mathbf{v},\mathbf{w}\in\mathbb{R}^{N_h}.\end{aligned}$$

Moreover, $\forall \mathbf{v}\in\mathbb{R}^{N_h}$,

$$(P_i\mathbf{v},\mathbf{v})_A = (AP_i\mathbf{v},\mathbf{v}) = (R_i^TA_i^{-1}R_iA\mathbf{v},A\mathbf{v}) = (A_i^{-1}R_iA\mathbf{v},R_iA\mathbf{v}) \geq 0. \qquad \diamond$$

Lemma 18.4. *The preconditioned matrix Q_a of the additive Schwarz method is symmetric and positive definite w.r.t the scalar product induced by A.*

Proof. Let us first prove the symmetry: for all $\mathbf{u},\mathbf{v}\in\mathbb{R}^{N_h}$, since A and P_i are both symmetric, we obtain

$$\begin{aligned}(Q_a\mathbf{u},\mathbf{v})_A &= (AQ_a\mathbf{u},\mathbf{v}) = (Q_a\mathbf{u},A\mathbf{v}) = \sum_i (P_i\mathbf{u},A\mathbf{v})\\ &= \sum_i (P_i\mathbf{u},\mathbf{v})_A = \sum_i (\mathbf{u},P_i\mathbf{v})_A = (\mathbf{u},Q_a\mathbf{v})_A.\end{aligned}$$

Concerning the positivity, choosing in the former identities $\mathbf{u}=\mathbf{v}$, we obtain

$$(Q_a\mathbf{v},\mathbf{v})_A = \sum_i (P_i\mathbf{v},\mathbf{v})_A = \sum_i (R_i^TA_i^{-1}R_iA\mathbf{v},A\mathbf{v}) = \sum_i (A_i^{-1}\mathbf{q}_i,\mathbf{q}_i) \geq 0,$$

having set $\mathbf{q}_i = R_iA\mathbf{v}$. It follows that $(Q_a\mathbf{v},\mathbf{v})_A = 0$ iff $\mathbf{q}_i=\mathbf{0}$ for every i, that is iff $A\mathbf{v}=\mathbf{0}$. Since A is positive definite, this holds iff $\mathbf{v}=\mathbf{0}$. $\diamond$

Owing to the previous properties we can deduce that a more efficient iterative method can be generated by replacing the preconditioned Richardson iterations with the preconditioned conjugate gradient iterations, yet using the same additive Schwarz

preconditioner P_{as}. Unfortunately, this preconditioner is not scalable. In fact, the condition number of the preconditioned matrix Q_a can only be bounded as

$$K_2(P_{as}^{-1}A) \leq C\frac{1}{\delta H}, \tag{18.122}$$

being C a constant independent of h, H and δ; here δ is a characteristic linear measure of the overlapping regions and, as usual, $H = \max_{i=1,\ldots,M}\{\text{diam}(\Omega_i)\}$. This is due to the fact that the exchange of information only occurs among neighbooring subdomains, as the application of $(P_{as})^{-1}$ involves only local solvers. This limitation can be overcome by introducing, also in the current context, a global coarse solver defined on the whole domain Ω and apt at guaranteing a global communication among all of the subdomains. This leads to devise two-level domain decomposition strategies, see Sect. 18.6.3.

Let us address some algorithmic aspects. Let us subdivide the domain Ω in M subdomains $\{\Omega_i\}_{i=1}^M$ such that $\cup_{i=1}^M \overline{\Omega}_i = \overline{\Omega}$. Neighbooring subdomains share an overlapping region of size at least equal to $\delta = \xi h$, for a suitable $\xi \in \mathbb{N}$. In particular, $\xi = 1$ corresponds to the case of minimum overlap, that is the overlapping strip reduces to a single layer of finite elements. The following algorithm can be used.

Algorithm 18.5 (introduction of overlapping subdomains)

a. Build a triangulation $\mathcal{T}_h$ of the computational domain Ω
b. Subdivide $\mathcal{T}_h$ in M disjoint subdomains $\{\hat{\Omega}_i\}_{i=1}^M$ such that $\cup_{i=1}^M \overline{\hat{\Omega}}_i = \overline{\Omega}$
c. Extend every subdomain $\hat{\Omega}_i$ by adding all the strips of finite elements of $\mathcal{T}_h$ within a distance δ from $\hat{\Omega}_i$. These extended subdomains identify the family of overlapping subdomains Ω_i

In Fig. 18.14 a rectangular two-dimensional domain is subdivided into 9 disjoint subdomains $\hat{\Omega}_i$ (on the left); also shown is one of the extended (overlapping) subdomains (on the right).

To apply the Schwarz preconditioner (18.120) we can proceed as indicated in **Algorithm 18.5**. We recall that N_i is the number of internal nodes of Ω_i, R_i^T and R_i are the prolongation and restriction matrices, respectively, introduced in (18.112) and A_i are

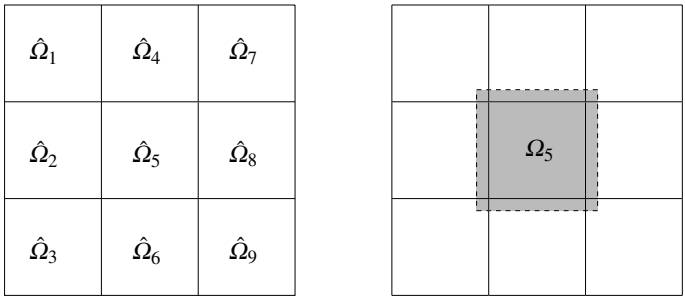

Fig. 18.14. Partition of a rectangular region Ω in 9 disjoint subregions $\hat{\Omega}_i$ (on the left), and an example of an extended subdomain Ω_5 (on the right)

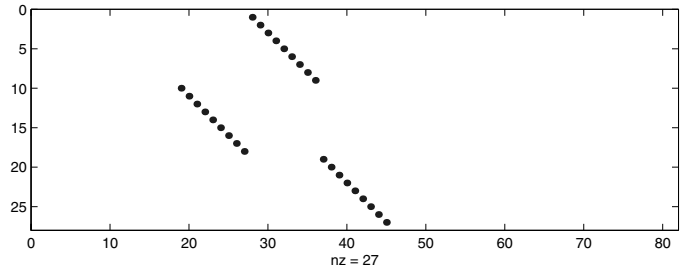

Fig. 18.15. The sparsity pattern of the matrix R_i for a partition of the domain in 4 subdomains

the local stiffness matrices introduced in (18.111). In Fig. 18.15 we display an example of sparsity pattern of R_i.

Algorithm 18.6 (start-up phase for the application of P_{as})

a. Build on every subdomanin Ω_i the matrices R_i and R_i^T

b. Build the stiffness matrix A corresponding to the finite element discretization on the grid $\mathcal{T}_h$

c. On every Ω_i build the local submatrices $A_i = R_i A R_i^T$

d. On every Ω_i set up the code for the solution of a linear system with matrix A_i.
For instance, compute a suitable (exact or incomplete) LU or Cholesky factorization of A_i

A few general comments on **Algorithm 18.5** and **Algorithm 18.6** are in order:

- steps a. and b. of algorithm 18.5 can be carried out in reverse order, that is we could first subdivide the computational domain into subdomains (based, for instance, on physical considerations), then set up a triangulation;
- depending upon the general code structure, steps b. and c. of the algorithm 18.6 could be glued together with the scope of optimizing memory requirements and CPU time.

In other circumstances we could interchange steps b. and c., that is the local stiffness matrices A_i can be built at first (using the single processors), then assembled to construct the global stiffness matrix A.
Indeed, a crucial factor for an efficient use of a parallel computer platform is keeping data locality since in most cases the time necessary for moving data among processors can be higher than that needed for computation.
Other codes (e.g. AztecOO, Trilinos, IFPACK) instead move from the global stiffness matrix distributed rowise and deduce the local stiffness matrices A_i without performing matrix-matrix products but simply using the column indices. In MATLAB, however, it seems more convenient to build A at first, next the restriction matrices R_i, and finally to carry out matrix multiplications $R_i A R_i^T$ to generate the A_i.

Table 18.4. Condition number of $P_{as}^{-1}A$ for several values of h and H

$K_2(P_{as}^{-1}A)$	$H = 1/2$	$H = 1/4$	$H = 1/8$	$H = 1/16$
$h = 1/16$	15.95	27.09	52.08	–
$h = 1/32$	31.69	54.52	104.85	207.67
$h = 1/64$	63.98	109.22	210.07	416.09
$h = 1/128$	127.99	218.48	420.04	832.57

In Table 18.4 we analyze the case of a decomposition with minimum overlap ($\delta = h$), considering several values for the number M of subdomains. The subdomains Ω_i are overlapping squares of area H^2. Note that the theoretical estimate (18.122) is satisfied by our results.

18.6.3 Two-level Schwarz preconditioners

As anticipated in Sect. 18.6.2, the main limitation of Schwarz methods is to propagate information only among neighbooring subdomains. As for the Neumann-Neumann method, a possible remedy consists of introducing a coarse grid mechanism that allows for a sudden information diffusion on the whole domain Ω. The idea is still that of considering the subdomains as macro-elements of a new coarse grid $\mathcal{T}_H$ and to build a corresponding stiffness matrix A_H. The matrix

$$Q_H = R_H^T A_H^{-1} R_H,$$

where R_H is the restriction operator from the fine to the coarse grid, represents the coarse level correction for the new two-level preconditioner. More precisely, setting for notational convenience $Q_0 = Q_H$, the two-level preconditioner P_{cas} is defined through its inverse as

$$P_{cas}^{-1} = \sum_{i=0}^{M} Q_i. \tag{18.123}$$

The following result can be proven in 2D: there exists a constant $C > 0$, independent of both h and H, such that

$$K_2(P_{cas}^{-1}A) \leq C(1 + \frac{H}{\delta}).$$

The ratio H/δ measures the relative overlap between neighboring overlapping subdomains. For "generous" overlap, that is if δ is a fraction of H, the preconditioner P_{cas} is scalable. Consequently, conjugate gradient iterations on the original finite element system using the preconditioner P_{cas} converges with a rate independent of h and H (and therefore of the number of subdomains). Moreover, thanks to the additive structure (18.123), the preconditioning step is fully parallel as it involves the solution of M independent systems, one per each local matrix A_i.

In 3D, we would get a bound with a factor H/h, unless the elliptic differential operator has constant coefficients (or variable coefficients which don't vary too much).

The use of P_{cas} involves the same kind of operations required by P_{as}, plus those of the following algorithm.

Algorithm 18.7 (start-up phase for the use of P_{cas})

a. Execute **Algorithm 18.6**

b. Define a coarse level triangulation $\mathcal{T}_H$ whose elements are of the order of H, then set $n_0 = \dim(V_0)$. Suppose that $\mathcal{T}_h$ be nested in $\mathcal{T}_H$. (See Fig. 18.16 for an example.)

c. Build the restriction matrix $R_0 \in \mathbb{R}^{n_0 \times N_h}$ whose elements are

$$R_0(i,j) = \Phi_i(\mathbf{x}_j),$$

where Φ_i is the basis function associated to the node i of the coarse grid, while by $\mathbf{x}_j$ we indicate the coordinates of the $j-th$ node on the fine grid

d. Build the coarse matrix A_H. This can be done by discretizing the original problem on the coarse grid $\mathcal{T}_H$, that is by computing

$$A_H(i,j) = a(\Phi_j, \Phi_i) = \int_\Omega \sum_{\ell=1}^{d} \frac{\partial \Phi_i}{\partial x_\ell} \frac{\partial \Phi_j}{\partial x_\ell},$$

or, otherwise, by setting

$$A_H = R_H A R_H^T.$$

For a computational domain with a simple shape (like the one we are considering) one typically generates the coarse grid $\mathcal{T}_H$ first, and then, by multiple refinements, the fine grid $\mathcal{T}_h$. In other cases, when the domain has a complex shape and/or a non struc-

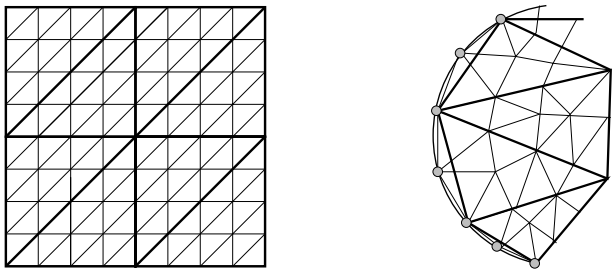

Fig. 18.16. On the left, example of a coarse grid for a 2D domain, based on a structured mesh. The triangles of the fine grid has thin edges; thick edges identify the boundaries of the coarse grid elements. On the right, a similar construction is displayed, this time for an unstructured fine grid

tured fine grid $\mathcal{T}_h$ is already available, the generation of a coarse grid might be difficult or computationally expensive. A first option would be to generate $\mathcal{T}_H$ by successive derefinements of the fine grid, in which case the nodes of the coarse grid will represent a subset of those of the fine grid. This approach, however, might not be very efficient in 3D.
Alternatively, one could generate the two (not necessarily nested) grids $\mathcal{T}_h$ and $\mathcal{T}_H$ independently, then generate the corresponding restriction and prolongation operators from the fine to the coarse grid, R_H and R_H^T.

The final implementation of P_{cas} could therefore be made as follows:

Algorithm 18.8 (P_{cas} solve)

For any given vector $\mathbf{r}$, the computation of $\mathbf{z} = P_{cas}^{-1}\mathbf{r}$ can be carried out as follows:

a. Set $\mathbf{z} = \mathbf{0}$

b. For $i = 1, \ldots, M$ Do in parallel:

c. restrict the residue on Ω_i: $\mathbf{r}_i = R_i\mathbf{r}$

d. compute $\mathbf{z}_i : A_i\mathbf{z}_i = \mathbf{r}_i$

e. add to the global residue: $\mathbf{z} \leftarrow R_i^T\mathbf{z}_i$

f. EndFor

g. Compute the coarse grid contribution $\mathbf{z}_H : A_H\mathbf{z}_H = R_H\mathbf{r}$

h. Add to the global residue: $\mathbf{z} \leftarrow R_H^T\mathbf{z}_H$

In Table 18.5 we report the condition number of $P_{cas}^{-1}A$ in the case of a minimum overlap $\delta = h$. Note that the condition number is almost the same on each NW-SE diagonal (i.e. for fixed values of the ratio H/δ).

An alternative approach to the coarse grid correction can be devised as follows. Suppose that the coefficients of the restriction matrix be given by

$$\hat{R}_H(i,j) = \begin{cases} 1 & \text{if the } j-th \text{ node is in } \Omega_i, \\ 0 & \text{otherwise,} \end{cases}$$

Table 18.5. Condition number of $P_{cas}^{-1}A$ for several values of h and H

$K_2(P_{cas}^{-1}A)$	$H = 1/4$	$H = 1/8$	$H = 1/16$	$H = 1/32$
$h = 1/32$	7.03	4.94	—	—
$h = 1/64$	12.73	7.59	4.98	—
$h = 1/128$	23.62	13.17	7.66	4.99
$h = 1/256$	45.33	24.34	13.28	—

Table 18.6. Condition number of $P_{aggre}^{-1}A$ for several values of h and H

$P_{aggre}^{-1}A$	$H = 1/4$	$H = 1/8$	$H = 1/16$
$h = 1/16$	13.37	8.87	—
$h = 1/32$	26.93	17.71	9.82
$h = 1/64$	54.33	35.21	19.70
$h = 1/128$	109.39	70.22	39.07

then we set $\hat{A}_H = \hat{R}_H A \hat{R}_H^T$. This procedure is named *aggregation* because the elements of $\hat{A}_H$ are obtained by simply summing up the entries of A. Note that we don't need to construct a coarse grid in this case. The corresponding preconditioner, denoted by P_{aggre}, has an inverse that reads

$$P_{aggre}^{-1} = \hat{R}_H^T \hat{A}_H^{-1} \hat{R}_H + P_{as}.$$

It can be proven that

$$K_2(P_{aggre}^{-1}A) \leq C \left(1 + \frac{H}{\delta}\right).$$

In Table 18.6 we report several numerical values of the condition number for different values of h and H.

If H/δ =constant, this two-level preconditioner is either optimal and scalable, that is the condition number of the preconditioned stiffness matrix is independent of both h and H.

We can conclude this section with the following practical indications:

- for decompositions with a small number of subdomains, the single level Schwarz preconditioner P_{as} is very efficient;
- when the number M of subdomains gets large, using two-level preconditioners becomes crucial; aggregation techniques can be adopted, in alternative to the use of a coarse grid in those cases in which the generation of the latter is difficult.

18.7 An abstract convergence result

The analysis of overlapping and non-overlapping domain decomposition preconditioners is based on the following abstract theory, due to P.L. Lions, J. Bramble, M. Dryja, O. Wildlund.

Let V_h be a Hilbert space of finite dimension. In our applications, V_h is one of the finite element spaces or spectral element spaces. Let V_h be decomposed as follows:

$$V_h = V_0 + V_1 + \cdots + V_M.$$

Let $F \in V'$ and $a : V \times V \to \mathbb{R}$ be a symmetric, continuous and coercive bilinear form.

Consider the problem

$$\text{find } u_h \in V_h : a(u_h, v_h) = F(v_h) \quad \forall v_h \in V_h. \tag{18.124}$$

Let $P_i : V_h \to V_i$ be a projection operator defined by

$$b_i(P_i u_h, v_h) = a(u_h, v_h) \quad \forall v_h \in V_i$$

with $b_i : V_i \times V_i \to \mathbb{R}$ being a local symmetric, continuous and coercive bilinear form on each subspace V_i. Assume that the following properties hold:

a. stable subspace decomposition:
$\exists C_0 > 0$ such that every $u_h \in V_h$ admits a decomposition $u_h = \sum_{i=0}^{M} u_i$ with $u_i \in V_i$ and

$$\sum_{i=0}^{M} b_i(u_i, u_i) \leq C_0^2 a(u_h, u_h);$$

b. strengthened Cauchy-Schwarz inequality:
$\exists \varepsilon_{ij} \in [0,1]$, $i, j = 0, \ldots, M$ such that

$$a(u_i, u_i) \leq \varepsilon_{ij} \sqrt{a(u_i, u_i)} \sqrt{a(u_j, u_j)} \quad \forall u_i \in V_i, u_j \in V_j;$$

c. local stability:
$\exists \omega \geq 1$ such that $\forall i = 0, \ldots, M$

$$a(u_i, u_i) \leq \omega b_i(u_i, u_i) \quad \forall u_i \in Range(P_i) \subset V_i.$$

Then, $\forall u_h \in V_h$,

$$C_0^{-2} a(u_h, u_h) \leq a(P_{as} u_h, u_h) \leq \omega(\rho(E) + 1) a(u_h, u_h) \tag{18.125}$$

where $\rho(E)$ is the spectral radius of the matrix $E = (\varepsilon_{ij})$, and $P_{as} = P_0 + \cdots + P_M$ is the domain decomposition preconditioner.
From inequality (18.125) the following bound holds for the preconditioned system

$$K(B^{-1}A) \leq C_0^2 \omega(\rho(E) + 1)$$

where $K(\cdot)$ denotes the spectral condition number, A the matrix associated with the original system (18.124), B the matrix associated to the operator P_{as}. For the proof, see e.g. [TW05].

18.8 Interface conditions for other differential problems

Theorem 18.1 in Sect. 18.1.2 allows a second order elliptic problem (18.1) to be reformulated in a DD version thanks to suitable interface conditions (18.9) and (18.10). On the other hand, as we have extensively discussed, such reformulation sets the ground for several iterative algorithms on disjoint DD partitions. They comprise Dirichlet-Neumann, Neumann-Neumann, Robin-Robin algorithms and, more generally, all of

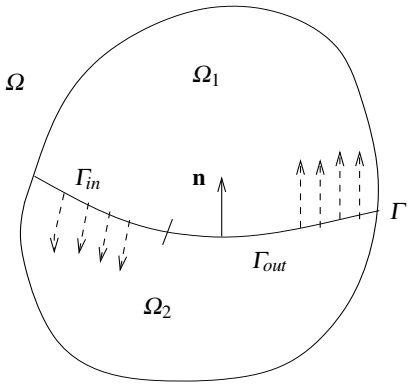

Fig. 18.17. Domain partition and interface splitting for the advection problem (18.126)

the preconditioned iterative algorithms of the Schur complement system (18.53) using suitable DD preconditioners.
In this section we consider other kind of boundary-value problems and formulate the associated interface conditions. Table 18.7 displays the interface conditions for these problems. For more details, analysis and investigation of associated iterative DD algorithms, the interested reader can consult [QV99].

Here we limit ourselves to provide a few additional insights in the case of advection and Stokes equations.

Advection (transport) problems. Consider the differential problem

$$Lu = \nabla \cdot (\mathbf{b}u) + a_0 u = f \qquad \text{in } \Omega, \tag{18.126}$$

supplemented by suitable conditions on the boundary $\partial\Omega$. Consider a partition of the computational domain Ω into two disjoint subdomains whose interface is Γ. Let us partition the latter as follows (see Fig. 18.17): $\Gamma = \Gamma_{in} \cup \Gamma_{out}$, where

$$\Gamma_{in} = \{x \in \Gamma |\ \mathbf{b}(x) \cdot \mathbf{n}(x) > 0\ \} \text{ and } \Gamma_{out} = \Gamma \setminus \Gamma_{in}.$$

Example 18.4. The Dirichlet-Neumann method for the problem at hand could be generalized as follows: being given two functions $u_1^{(0)}$, $u_2^{(0)}$ on Γ, $\forall k \geq 0$ solve:

$$\begin{cases} Lu_1^{(k+1)} = f & \text{in } \Omega_1, \\ (\mathbf{b} \cdot \mathbf{n}) u_1^{(k+1)} = (\mathbf{b} \cdot \mathbf{n}) u_2^{(k)} & \text{on } \Gamma_{out}, \end{cases}$$

$$\begin{cases} Lu_2^{(k+1)} = f & \text{in } \Omega_2, \\ (\mathbf{b} \cdot \mathbf{n}) u_2^{(k+1)} = \theta (\mathbf{b} \cdot \mathbf{n}) u_1^{(k)} + (1-\theta)(\mathbf{b} \cdot \mathbf{n}) u_2^{(k)} & \text{on } \Gamma_{in}. \end{cases}$$

where $\theta > 0$ denotes a suitable relaxation parameter. The adaptation to the case of a finite element discretization is straightforward. ■

Stokes problem. The Stokes equations (16.11) feature two fields of variables: fluid velocity and fluid pressure. When considering a DD partition, at subdomain interface only the velocity field is requested to be continuous. Pressure needs not necessarily be continuous, since in the weak formulation of the Stokes equations it is "only" requested to be in L^2. Moreover, on the interface Γ the continuity of the normal Cauchy stress $\nu\frac{\partial \mathbf{u}}{\partial n} - p\mathbf{n}$ needs only be satisfied in weak (natural) form.

Example 18.5. A Dirichlet-Neumann algorithm for the Stokes problem would entail at each iteration the solution of the following subproblems (we use the short-hand notation $\mathcal{S}$ to indicate the Stokes operator):

$$\begin{cases} \mathcal{S}(\mathbf{u}_2^{(k+1)}, p_2^{(k+1)}) = \mathbf{f} & \text{in } \Omega_2, \\ \nu\dfrac{\partial \mathbf{u}_2^{(k+1)}}{\partial n} - p_2^{(k+1)} = \nu\dfrac{\partial \mathbf{u}_1^{(k)}}{\partial n} - p_1^{(k)} & \text{on } \Gamma, \\ \mathbf{u}_2^{(k+1)} = \mathbf{0} & \text{on } \partial\Omega_2 \setminus \Gamma, \end{cases} \tag{18.127}$$

$$\begin{cases} \mathcal{S}(\mathbf{u}_1^{(k+1)}, p_1^{(k+1)}) = \mathbf{f} & \text{in } \Omega_1, \\ \mathbf{u}_1^{(k+1)} = \theta\mathbf{u}_2^{(k+1)} + (1-\theta)\mathbf{u}_1^{(k)} & \text{on } \Gamma, \\ \mathbf{u}_1^{(k+1)} = \mathbf{0} & \text{on } \partial\Omega_1 \setminus \Gamma. \end{cases} \tag{18.128}$$

Should the boundary conditions of the original problem be prescribed on the velocity field, e.g. $\mathbf{u} = \mathbf{0}$, pressure p would be defined only up to an additive constant, which could be fixed by, e.g., imposing the constraint $\int_\Omega p\, d\Omega = 0$.
To fulfill this constraint we can proceed as follows. When solving the Neumann problem (18.127) on the subdomain Ω_2, both the velocity $\mathbf{u}_2^{(k+1)}$ and the pressure $p_2^{(k+1)}$ are univocally determined. When solving the Dirichlet problem (18.128) on Ω_1, the pressure is defined only up to an additive constant; we fix it by imposing the additional equation

$$\int_{\Omega_1} p_1^{(k+1)}\, d\Omega_1 = -\int_{\Omega_2} p_2^{(k+1)}\, d\Omega_2.$$

Should the four sequences $\{\mathbf{u}_1^{(k)}\}$, $\{\mathbf{u}_2^{(k)}\}$, $\{p_1^{(k)}\}$ and $\{p_2^{(k)}\}$ converge, the null average condition on the pressure would be automatically verified. ■

Example 18.6. Suppose now that the Schwarz iterative method is used on an overlapping subdomain decomposition of the domain like that on Fig. 18.1, left. At every step we have to solve two Dirichlet problems for the Stokes equations:

$$\begin{cases} \mathcal{S}(\mathbf{u}_1^{(k+1)}, p_1^{(k+1)}) = \mathbf{f} & \text{in } \Omega_1, \\ \mathbf{u}_1^{(k+1)} = \mathbf{u}_2^{(k)} & \text{on } \Gamma_1, \\ \mathbf{u}_1^{(k+1)} = 0 & \text{on } \partial\Omega_1 \setminus \Gamma_1, \end{cases} \tag{18.129}$$

$$\begin{cases} \mathcal{S}(\mathbf{u}_2^{(k+1)}, p_2^{(k+1)}) = \mathbf{f} & \text{in } \Omega_2, \\ \mathbf{u}_2^{(k+1)} = \mathbf{u}_1^{(k+1)} & \text{on } \Gamma_2, \\ \mathbf{u}_2^{(k+1)} = 0 & \text{on } \partial\Omega_2 \setminus \Gamma_2. \end{cases} \tag{18.130}$$

No continuity is required on the pressure field at subdomain boundaries.

The constraint on the fluid velocity to be divergence free on the whole domain Ω requires special care. Indeed, after solving (18.129), we have $\mathrm{div}\mathbf{u}_1^{(k+1)} = 0$ in Ω_1, hence, thanks to the Green formula,

$$\int_{\partial\Omega_1} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n}\, d\gamma = 0.$$

This relation implies a similar relation for $\mathbf{u}_2^{(k)}$ in $(18.129)_2$; indeed

$$0 = \int_{\partial\Omega_1} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n}\, d\gamma = \int_{\Gamma_1} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n}\, d\gamma = \int_{\Gamma_1} \mathbf{u}_2^{(k)} \cdot \mathbf{n}\, d\gamma. \tag{18.131}$$

At the very first iteration we can select $\mathbf{u}_2^{(0)}$ in such a way that the compatibility condition (18.131) be satisfied, however this control is lost, a priori, in the course of the subsequent iterations. For the same reason, the solution of (18.130) yields the compatibility condition

$$\int_{\Gamma_2} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n}\, d\gamma = 0. \tag{18.132}$$

Fortunately, Schwarz method automatically guarantees that this condition holds. Indeed, in $\Gamma_{12} = \Omega_1 \cap \Omega_2$ we have $\mathrm{div}\mathbf{u}_1^{(k+1)} = 0$, moreover on $\Gamma_{12} \setminus (\Gamma_1 \cup \Gamma_2)$, $\mathbf{u}_1^{(k+1)} = \mathbf{0}$ because of the given homogeneous Dirichlet boundary conditions. Thus

$$0 = \int_{\partial\Gamma_{12}} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n}\, d\gamma = \int_{\Gamma_1} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n}\, d\gamma + \int_{\Gamma_2} \mathbf{u}_1^{(k+1)} \cdot \mathbf{n}\, d\gamma.$$

The first integral on the right hand side vanishes because of (18.131), therefore (18.132) is satisfied. ■

18.9 Exercises

1. Consider the one-dimensional advection-transport-reaction problem

$$\begin{cases} -(\alpha u_x)_x + (\beta u)_x + \gamma u = f & \text{in } \Omega = (a,b) \\ u(a) = 0, \quad \alpha u_x(b) - \beta u(b) = g, \end{cases} \tag{18.133}$$

with α and $\gamma \in \mathrm{L}^\infty(a,b)$, $\beta \in \mathrm{W}^{1,\infty}(a,b)$ and $f \in L^2(a,b)$.

a) Write the additive Schwarz iterative method, then the multiplicative one, on the two overlapping intervals $\Omega_1 = (a, \gamma_2)$ and $\Omega_2 = (\gamma_1, b)$, with $a < \gamma_1 < \gamma_2 < b$.

Table 18.7. Interface continuity conditions for several kind of differential operators; D stands for Dirichlet condition, N for Neumann

Operator	*Problem*	*D*	*N*
Laplace	$-\triangle u = f,$	u	$\dfrac{\partial u}{\partial n}$
Elasticity	$-\nabla\cdot(\sigma(\mathbf{u})) = \mathbf{f},$ with $\sigma_{kj} = \hat{\mu}(D_k u_j + D_j u_k) + \hat{\lambda}\,\mathrm{div}\mathbf{u}\delta_{kj}\,,$ $\mathbf{u}$ in-plane membrane displacement	$\mathbf{u}$	$\sigma(\mathbf{u})\cdot\mathbf{n}$
Transport-diffusion	$-\sum_{kj} D_k(A_{kj}D_j u) + \mathrm{div}(\mathbf{b}u) + a_0 u = f$	u	$\dfrac{\partial u}{\partial n_L} = \sum_k a_{kj} D_j u \cdot n_k$
Transport	$\mathrm{div}(\mathbf{b}u) + a_0 u = f$		$\mathbf{b}\cdot\mathbf{n}\,u$
Incompressible viscous flows	$-\mathrm{div}\mathsf{T}(\mathbf{u},p) + (\mathbf{u}^*\cdot\nabla)\mathbf{u} = \mathbf{f},$ $\mathrm{div}\mathbf{u} = 0,$ with $\mathsf{T}_{kj} = \nu(D_k u_j + D_j u_k) - p\delta_{kj},$ $\mathbf{u}^* = \begin{cases} 0 & \text{(Stokes equations)} \\ \mathbf{u}_\infty & \text{(Oseen equations)} \\ \mathbf{u} & \text{(Navier-Stokes equations)} \end{cases}$	$\mathbf{u}$	$\mathsf{T}(\mathbf{u},p)\cdot\mathbf{n}$
Compressible viscous flows	$\alpha\mathbf{u} - \mathrm{div}\hat{\mathsf{T}}(\mathbf{u},\sigma) = \mathbf{f},$ $\alpha\sigma + \mathrm{div}\mathbf{u} = g,$ with $\hat{\mathsf{T}}_{kj} = \nu(D_k u_j + D_j u_k)$ $-\beta\sigma\delta_{kj} + \left(g - \frac{2\nu}{d}\right)\mathrm{div}\mathbf{u}\delta_{kj},$ $\rho = \text{fluid density} = \log\sigma$	$\mathbf{u}$	$\hat{\mathsf{T}}(\mathbf{u},\sigma)\cdot\mathbf{n}$
Compressible inviscid flows	$\alpha\mathbf{u} + \beta\nabla\sigma = \mathbf{f},$ $\alpha\sigma + \mathrm{div}\mathbf{u} = 0$	$\mathbf{u}\cdot\mathbf{n}$	σ
Maxwell (harmonic regime)	$\mathbf{rot}\left(\dfrac{1}{\mu}\mathbf{rotE}\right)$ $-\alpha^2\varepsilon\mathbf{E} + i\alpha\sigma\mathbf{E} = \mathbf{f}$	$\mathbf{n}\times\mathbf{E}$	$\mathbf{n}\times\left(\dfrac{1}{\mu}\mathbf{rotE}\right)$

b) Interpret these methods as suitable Richardson algorithms to solve the given differential problem.
c) In case we approximate (18.133) by the finite element method, write the corresponding additive Schwarz preconditioner, with and without coarse-grid component. Then provide an estimate of the condition number of the preconditioned matrix, in both cases.

2. Consider the one-dimensional diffusion-transport-reaction problem

$$\begin{cases} -(\alpha u_x)_x + (\beta u)_x + \delta u = f & \text{in } \Omega = (a,b) \\ \alpha u_x(a) - \beta u(a) = g, \quad u_x(b) = 0, \end{cases} \tag{18.134}$$

with α and $\gamma \in \mathrm{L}^\infty(a,b)$, $\alpha(x) \geq \alpha_0 > 0$, $\beta \in \mathrm{W}^{1,\infty}(a,b)$, $f \in L^2(a,b)$ and g a given real number.

a) Consider two disjoined subdomains of Ω, $\Omega_1 = (a,\gamma)$ and $\Omega_2 = (\gamma,b)$, with $a < \gamma < b$. Formulate problem (18.134) using the Steklov-Poincaré operator, both in differential and variational form. Analyze the properties of this operator starting from those of the bilinear form associated with problem (18.134).
b) Apply the Dirichlet-Neumann method to problem (18.134) using the same domain partition introduced at point a).
c) In case of finite element approximation, derive the expression of the Dirichlet-Neumann preconditioner of the Schur complement matrix.

3. Consider the one-dimensional Poisson problem

$$\begin{cases} -u_{xx}(x) = f(x) & \text{in } \Omega = (0,1) \\ u(0) = 0, \quad u_x(1) = 0, \end{cases} \tag{18.135}$$

with $f \in L^2(\Omega)$.

a) If $\mathcal{T}_h$ indicates a partition of the interval Ω with step-size h, write the Galerkin-finite element approximation of problem (18.135).
b) Consider now a partition of Ω into the subintervals $\Omega_1 = (0,\gamma)$ and $\Omega_2 = (\gamma,1)$, being $0 < \gamma < 1$ a node of the partition $\mathcal{T}_h$ (See Fig. 18.18). Write the algebraic blockwise form of the Galerkin-finite element stiffness matrix relative to this subdomain partition.
c) Derive the discrete Steklov-Poincaré interface equation which corresponds to the DD formulation at point b). Which is the dimension of the Schur complement?
d) Consider now two overlapping subdomains $\Omega_1 = (0,\gamma_2)$ and $\Omega_2 = (\gamma_1,1)$, with $0 < \gamma_1 < \gamma_2 < 1$, the overlap being reduced to a single finite element of the

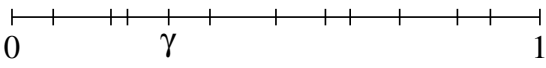

Fig. 18.18. Subdomain partition $\mathcal{T}_h$ of the interval $(0,1)$

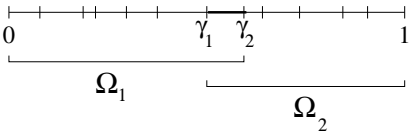

Fig. 18.19. Overlapping decomposition of the interval $(0,1)$

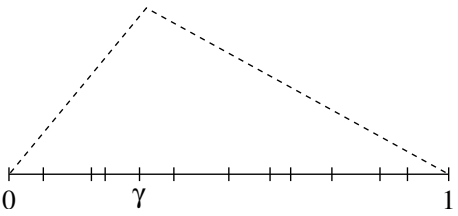

Fig. 18.20. Coarse-grid partition made of two macro elements for the construction of matrix A_H and Lagrangian characteristic function associated with the node γ

partition $\mathcal{T}_h$ (see Fig. 18.19). Provide the algebraic formulation of the additive Schwarz iterative method.

e) Provide the general expression of the two-level additive Shwarz preconditioner, by assuming as coarse matrix A_H that associated with only two elements, as displayed in Fig. 18.20.

4. Consider the diffusion-transport-reaction problem

$$\begin{cases} Lu = -\nabla\cdot(\alpha\nabla u) + \nabla\cdot(\beta u) + \gamma u = f & \text{in } \Omega = (0,2)\times(0,1), \\ u = 0 & \text{on } \Gamma_D, \\ \alpha\dfrac{\partial u}{\partial n} + \delta u = 0 & \text{on } \Gamma_R, \end{cases} \tag{18.136}$$

with $\alpha = \alpha(\mathbf{x})$, $\beta = \beta(\mathbf{x})$, $\gamma = \gamma(\mathbf{x})$, $\delta = \delta(\mathbf{x})$ and $f = f(\mathbf{x})$ being given functions, and $\partial\Omega = \overline{\Gamma}_D \cup \overline{\Gamma}_R$, with $\overset{\circ}{\Gamma}_D \cap \overset{\circ}{\Gamma}_R = \emptyset$.
Let Ω in (18.136) be partitioned into two disjoined subdomains $\Omega_1 = (0,1)\times(0,1)$ and $\Omega_2 = (1,2)\times(0,1)$.

a) Formulate problem (18.136) in terms of the Steklov-Poincaré operator, both in differential and variational form.

b) Apply the Dirichlet-Neumann method to problem (18.136) using the same decomposition introduced before.

c) Prove the equivalence between the Dirichlet-Neumann method at point b) and a suitable preconditioned Richardson operator, after setting $\alpha = 1$, $\beta = 0$, $\gamma = 1$ and $\Gamma_R = \emptyset$ in (18.136). Do the same for the Neumann-Neumann method.

5. Consider the two-dimensional diffusion-transport-reaction problem

$$\begin{cases} Lu = -\nabla \cdot (\mu \nabla u) + \boldsymbol{b} \cdot \nabla u + \sigma u = f & \text{in} \quad \Omega = (a,c) \times (d,e), \\ u = 0 & \text{on} \quad \partial\Omega. \end{cases} \tag{18.137}$$

Consider a decomposition of Ω made of the overlapping subdomains $\Omega_3 = (a,f) \times (d,e)$ and $\Omega_4 = (g,c) \times (d,e)$, with $g < f$. On such a decomposition, write for problem (18.137) the Schwarz method in both multiplicative and additive versions. Then interpret these methods as suitable preconditioned Richardson iterative algorithms. Finally, comment on the convergence properties of these methods.