Seven things I would have liked to know when starting to work on Domain Decomposition

Martin J. Gander[1]

University of Geneva, Geneva, 1211, Switzerland,
martin.gander@unige.ch,
WWW home page: https://www.unige.ch/~gander

Abstract. I give in this paper a very personal view on the active research field of Domain Decomposition methods. I explain why it was difficult for me to start working in the field, and how I managed, applying the technique of decomposition itself to handle the difficulty of studying domain decomposition methods, to develop nevertheless an understanding of the field, and loving it. Even though I have not yet converged to the limit of fully understanding the most powerful domain decomposition techniques combining all the pieces, I show for example how to modify two level Schwarz methods so that they are as effective to solve Poisson problems as multigrid in wallclock time, and this without Krylov acceleration. I hope sharing my experience will help to transmit the passion I developed for this field of computational and applied mathematics, and motivate readers to develop these techniques further.

Keywords: Domain Decomposition, Schwarz, Dirichlet-Neumann, Neumann-Neumann, FETI, Discretization, Coarse spaces, Krylov acceleration.

Introduction

"My name is Bond, James Bond", 007.

When I talked for the first time about the subject in the title in January 2018 at the Ecole Polytechnique in Paris, I was asked if I had explicitly tried to get to the number 7. Seven is a special number, that appears quite often in various contexts, like the 7 dwarfs, the 7 deadly sins, the 7 days of creation, we have 7 days in the week, there are 7 world wonders, there are 7 notes in the musical scale, and so on (see also the quote above for the more popular culture). In the present case however, it just happened to me that there were 7 things[1], when I thought about what I would like to tell the next generation of researchers and scientists that were interested in Domain Decomposition (DD) methods, or that were simply obliged to consider them because they are the only option for performing their simulation, in order to obtain a solution.

Starting to do research in an already mature field of research like DD methods is not easy, and the learning curve is very steep. This is because if one

[1] And now it took seven years to finalize this manuscript!

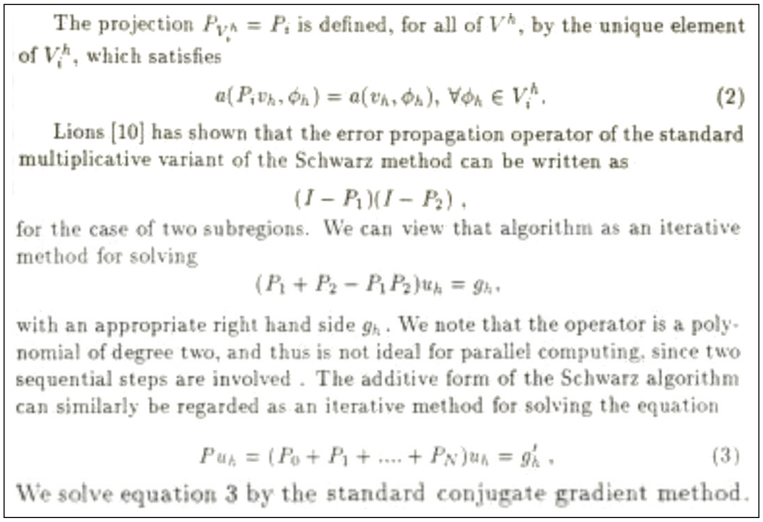
wants to understand the most complete and seminal results in DD methods, one needs deep knowledge in functional analysis, partial differential equations, their discretization, numerical linear algebra and iterative methods, computer science including hardware and software, programming, and furthermore also applications. For me it was not possible to start right away with these most complete and seminal results, I had to approach them with small steps which I was able to understand, disconnect the various difficulties by decomposing them, like in DD itself, and still I have not yet fully converged to the fundamental and complete understanding contained in these complete and seminal results.

I therefore start in the next section by introducing the four main seminal results in DD, for which the book [74] is the reference. I then decompose the difficulties I had for understanding them, and explain piece by piece how I managed to improve my understanding, getting closer and closer to currently active research questions of interest to me. In Section 2, I explain the classical four DD solvers as iterations at the continuous level for a simple model problem, without Krylov acceleration, and show why and how they converge. In Section 3, I show how such methods can be discretized and implemented on a computer. In Section 4, I introduce better transmission conditions than the classical Dirichlet and Neumann conditions used by the main four classical domain decomposition methods from Sections 1 and 2, and explain research directions these transmission conditions opened up. In Section 5, I explain the so important term 'optimal' used in DD, and propose as alternative 'scalable', which makes it easier to keep developing DD and also more general iterative methods, and encourages further research to enhance them. In Section 6, I illustrate how this helped me to investigate local and global convergence mechanisms separately and in combination, which leads to a new understanding of coarse spaces for such methods. And it is only at the end that I introduce Krylov acceleration of all these methods in Section 7, since Krylov methods can be considered as accelerators of the underlying DD iteration, and are difficult to integrate into the theory, except for the very special case of symmetric positive definite problems and the conjugate gradient method (CG), where one can argue with condition numbers. I make a few concluding remarks in the last section. Again, I would like to emphasize that this is a very personal and thus highly biased vision of the field, but I hope it is helpful, and also instructive and fun to read.

1 Seminal contributions to DD not easy to start with

> *"I worked hard. Anyone who works as hard as I did can achieve the same results."*
>
> Johann Sebastian Bach.

Modern DD methods are highly sophisticated parallel solvers for partial differential equations (PDEs), which combine many different techniques in the realm of iterative methods for solving large scale sparse linear systems. It can be daunting for young scientists that are new in the field to try to understand directly methods at this level of sophistication. I show in this section the main



The projection $P_{V_i^h} = P_i$ is defined, for all of V^h, by the unique element of V_i^h, which satisfies

$$a(P_i v_h, \phi_h) = a(v_h, \phi_h),\ \forall \phi_h \in V_i^h. \tag{2}$$

Lions [10] has shown that the error propagation operator of the standard multiplicative variant of the Schwarz method can be written as

$$(I - P_1)(I - P_2)\ ,$$

for the case of two subregions. We can view that algorithm as an iterative method for solving

$$(P_1 + P_2 - P_1 P_2) u_h = g_h ,$$

with an appropriate right hand side g_h . We note that the operator is a polynomial of degree two, and thus is not ideal for parallel computing, since two sequential steps are involved . The additive form of the Schwarz algorithm can similarly be regarded as an iterative method for solving the equation

$$P u_h = (P_0 + P_1 + ... + P_N) u_h = g_h' \ , \tag{3}$$

We solve equation 3 by the standard conjugate gradient method.

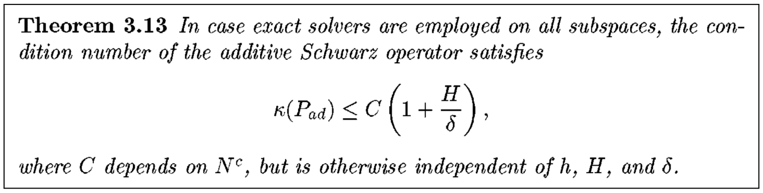
Fig. 1. Seminal invention of the Additive Schwarz Method.

four DD methods at their full level of sophistication, illustrated by their invention, and recall the main convergence results known about them. The first three Olof Widlund prizes in DD methods were won for three of these inventions.

1.1 The Additive Schwarz Method

In 1987, Max Dryja and Olof Widlund introduced in the technical report [28] what was to become one of the most used and analyzed DD methods in the field, namely the Additive Schwarz (AS) method. The method is defined for a given bilinear form in a Hilbert space with given subspaces of the Hilbert space, and only uses projections in its definition, no PDEs, see Fig. 1. As we can see from the quote, it was motivated by the analysis of Lions of Schwarz methods in the first of three seminal contributions [63], where Lions wrote the alternating Schwarz method as an alternating projection method in a Hilbert space. The main idea of Dryja and Widlund was to obtain a parallel version of this alternating projection method by executing the projections in parallel. In addition, the method is not considered as an iteration, but as a new system which is to be solved by CG, see the last sentence in Fig. 1.

In order to understand the properties of this new, so called preconditioned system given by the sum of the projections, the abstract Schwarz framework was developed, which led to the condition number estimate for AS shown in Fig. 2. The estimate states in very general terms that if the domain decomposition has subdomains with generic diameter H, and these subdomains overlap with an overlap size δ, and if the ratio of $\frac{H}{\delta}$ is constant, then the condition number remains bounded by a constant, and hence CG will converge in a number of iterations that is independent of the number of subdomains and the overlap δ

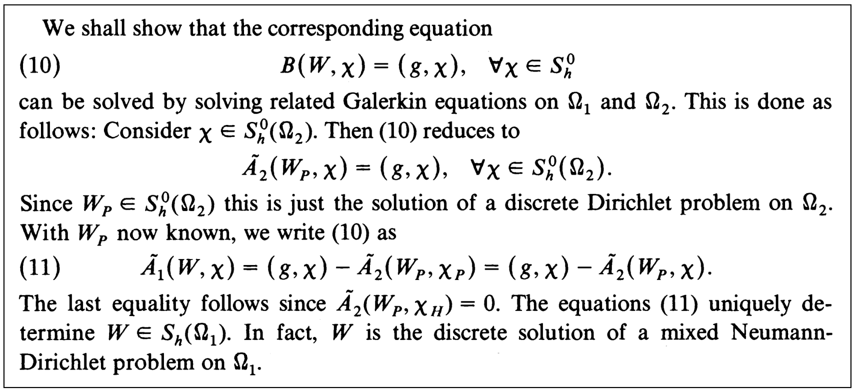


Theorem 3.13 *In case exact solvers are employed on all subspaces, the condition number of the additive Schwarz operator satisfies*

$$\kappa(P_{ad}) \leq C\left(1+\frac{H}{\delta}\right),$$

where C depends on N^c, but is otherwise independent of h, H, and δ.

Fig. 2. Fundamental condition number estimate for 2-level AS, taken from [74].



We shall show that the corresponding equation

$$(10) \qquad B(W,\chi) = (g,\chi), \quad \forall \chi \in S_h^0$$

can be solved by solving related Galerkin equations on Ω_1 and Ω_2. This is done as follows: Consider $\chi \in S_h^0(\Omega_2)$. Then (10) reduces to

$$\tilde{A}_2(W_P,\chi) = (g,\chi), \quad \forall \chi \in S_h^0(\Omega_2).$$

Since $W_P \in S_h^0(\Omega_2)$ this is just the solution of a discrete Dirichlet problem on Ω_2. With W_P now known, we write (10) as

$$(11) \qquad \tilde{A}_1(W,\chi) = (g,\chi) - \tilde{A}_2(W_P,\chi_P) = (g,\chi) - \tilde{A}_2(W_P,\chi).$$

The last equality follows since $\tilde{A}_2(W_P,\chi_H) = 0$. The equations (11) uniquely determine $W \in S_h(\Omega_1)$. In fact, W is the discrete solution of a mixed Neumann-Dirichlet problem on Ω_1.

Fig. 3. The appearence of the Dirichlet-Neumann method from [7].

often proportional to the mesh size h used to discretize the problem. Note that here the additive Schwarz operator also contains a coarse space.

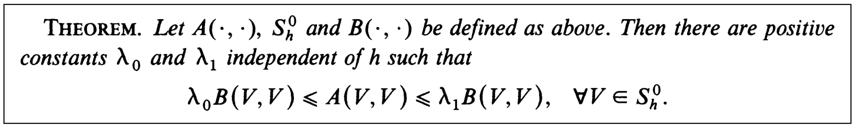
The convergence analysis techniques developed for AS have reached such a level of generality that the so called abstract Schwarz framework has also been used to analyze DD methods which are quite different from the original Schwarz method, as we will see in the following subsections.

1.2 The Dirichlet-Neumann Method

Around the same time as AS, the Dirichlet-Neumann method was introduced by Bramble, Pasciak and Schatz [7], see Figure 3, and also by Bjorstad and Widlund [4]; see also their earlier work [3] where they cite already a preprint of Bramble, Pasciak and Schatz. Like AS, the method was directly introduced as a preconditioner for CG at the discrete level, and a condition number estimate was proved for the case of two subdomains, see Figure 4, which shows that the method converges independently of the mesh size h.

1.3 The FETI Method

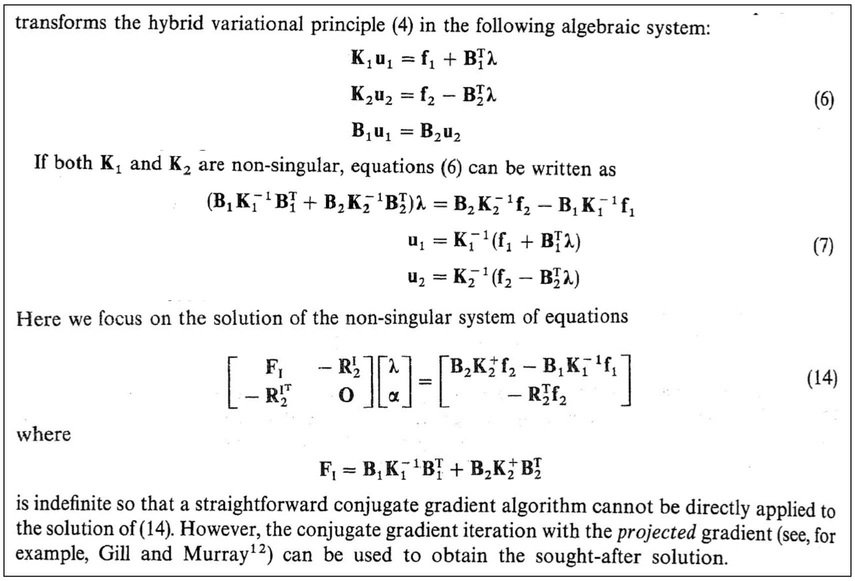
Four years after the seminal invention of AS, Farhat and Roux introduced in 1991 the Finite Element Tearing and Interconnect (FETI) DD method [31]. The



Theorem. *Let $A(\cdot,\cdot)$, S_h^0 and $B(\cdot,\cdot)$ be defined as above. Then there are positive constants λ_0 and λ_1 independent of h such that*

$$\lambda_0 B(V,V) \leqslant A(V,V) \leqslant \lambda_1 B(V,V), \quad \forall V \in S_h^0.$$

Fig. 4. Condition number estimate for the Dirichlet-Neumann method from [7].



transforms the hybrid variational principle (4) in the following algebraic system:

$$\begin{aligned} \mathbf{K}_1\mathbf{u}_1 &= \mathbf{f}_1 + \mathbf{B}_1^{\mathrm{T}}\lambda \\ \mathbf{K}_2\mathbf{u}_2 &= \mathbf{f}_2 - \mathbf{B}_2^{\mathrm{T}}\lambda \\ \mathbf{B}_1\mathbf{u}_1 &= \mathbf{B}_2\mathbf{u}_2 \end{aligned} \tag{6}$$

If both $\mathbf{K}_1$ and $\mathbf{K}_2$ are non-singular, equations (6) can be written as

$$\begin{aligned} (\mathbf{B}_1\mathbf{K}_1^{-1}\mathbf{B}_1^{\mathrm{T}} + \mathbf{B}_2\mathbf{K}_2^{-1}\mathbf{B}_2^{\mathrm{T}})\lambda &= \mathbf{B}_2\mathbf{K}_2^{-1}\mathbf{f}_2 - \mathbf{B}_1\mathbf{K}_1^{-1}\mathbf{f}_1 \\ \mathbf{u}_1 &= \mathbf{K}_1^{-1}(\mathbf{f}_1 + \mathbf{B}_1^{\mathrm{T}}\lambda) \\ \mathbf{u}_2 &= \mathbf{K}_2^{-1}(\mathbf{f}_2 - \mathbf{B}_2^{\mathrm{T}}\lambda) \end{aligned} \tag{7}$$

Here we focus on the solution of the non-singular system of equations

$$\begin{bmatrix} \mathbf{F}_{\mathrm{I}} & -\mathbf{R}_2^{\mathrm{I}} \\ -\mathbf{R}_2^{\mathrm{IT}} & \mathbf{O} \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{B}_2\mathbf{K}_2^{+}\mathbf{f}_2 - \mathbf{B}_1\mathbf{K}_1^{-1}\mathbf{f}_1 \\ -\mathbf{R}_2^{\mathrm{T}}\mathbf{f}_2 \end{bmatrix} \tag{14}$$

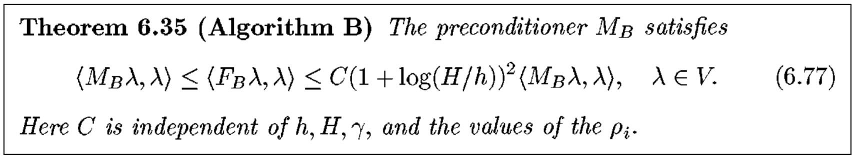
where

$$\mathbf{F}_{\mathrm{I}} = \mathbf{B}_1\mathbf{K}_1^{-1}\mathbf{B}_1^{\mathrm{T}} + \mathbf{B}_2\mathbf{K}_2^{+}\mathbf{B}_2^{\mathrm{T}}$$

is indefinite so that a straightforward conjugate gradient algorithm cannot be directly applied to the solution of (14). However, the conjugate gradient iteration with the *projected* gradient (see, for example, Gill and Murray[12]) can be used to obtain the sought-after solution.

Fig. 5. Seminal invention of the FETI method from [31].

method was invented using a minimization formulation of the partial differential equation to solve, and led to a linear system of equations containing so called Lagrange multipliers, see Fig. 5. Again there is no iteration, and the system is solved by CG, this time however with a projected gradients. The FETI method has become one of the most researched and thoroughly numerically tested DD method, also on very large scale industrial problems. Like for AS, there is a very general condition number estimate, as shown in Fig. 6. We see that the estimate is written a bit differently, but if one divides on both sides by the inner product $< M_B\lambda, \lambda >$ one gets a condition number estimate of the same form as for AS. Here however, one needs to keep the subdomain width H divided by the small mesh size h constant in order to get a method that converges with projected CG independently of the mesh size and number of subdomains. Note that also here there is a coarse space, indicated by the abbreviation DP (Dual-Primal), and there is also an additional preconditioner.

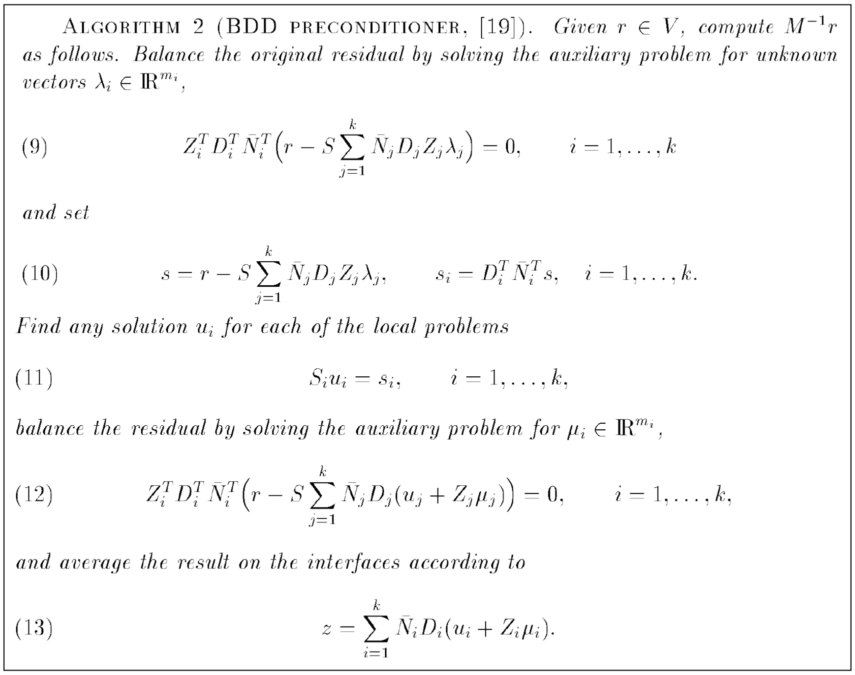


Theorem 6.35 (Algorithm B) *The preconditioner M_B satisfies*

$$\langle M_B\lambda,\lambda\rangle \leq \langle F_B\lambda,\lambda\rangle \leq C(1+\log(H/h))^2\langle M_B\lambda,\lambda\rangle, \quad \lambda\in V. \qquad (6.77)$$

Here C is independent of h, H, γ, and the values of the ρ_i.

Fig. 6. Fundamental condition number estimate for FETI-DP, taken from [74].



ALGORITHM 2 (BDD PRECONDITIONER, [19]). *Given $r \in V$, compute $M^{-1}r$ as follows. Balance the original residual by solving the auxiliary problem for unknown vectors $\lambda_i \in \mathrm{I\!R}^{m_i}$,*

$$(9) \qquad Z_i^T D_i^T \bar{N}_i^T\Big(r - S\sum_{j=1}^{k}\bar{N}_j D_j Z_j \lambda_j\Big) = 0, \qquad i = 1,\ldots,k$$

and set

$$(10) \qquad s = r - S\sum_{j=1}^{k}\bar{N}_j D_j Z_j\lambda_j, \qquad s_i = D_i^T\bar{N}_i^T s, \quad i = 1,\ldots,k.$$

Find any solution u_i for each of the local problems

$$(11) \qquad S_i u_i = s_i, \qquad i = 1,\ldots,k,$$

balance the residual by solving the auxiliary problem for $\mu_i \in \mathrm{I\!R}^{m_i}$,

$$(12) \qquad Z_i^T D_i^T \bar{N}_i^T\Big(r - S\sum_{j=1}^{k}\bar{N}_j D_j(u_j + Z_j\mu_j)\Big) = 0, \qquad i = 1,\ldots,k,$$

and average the result on the interfaces according to

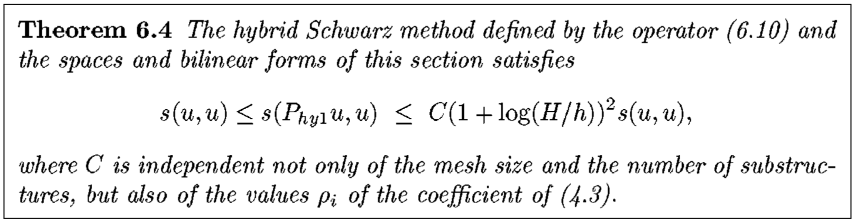
$$(13) \qquad z = \sum_{i=1}^{k}\bar{N}_i D_i(u_i + Z_i\mu_i).$$

Fig. 7. Seminal invention of the BDD method from [65].

1.4 The Balancing DD Method

Two years after the invention of FETI, Mandel and Brezina introduced in 1993 the Balancing DD method (BDD) in the technical report [65], see Fig. 7. The method is described at the algebraic level, by giving precisely the steps that have to be executed to apply the preconditioner to a residual. BDD methods are the fourth and last class of DD methods that have been thoroughly analyzed and tested, and there is also a condition number estimate, given in Fig. 8. There are two surprising things to note in this condition number estimate for BDD: the first one is that the theorem talks about a hybrid Schwarz method, and not about BDD, and the second is that the condition number estimate is essentially identical to the FETI-DP condition number estimate in Fig. 6!

When I started to work on DD methods about 30 years ago, I was going to conferences and studied the corresponding literature, and it was extremely



Theorem 6.4 *The hybrid Schwarz method defined by the operator (6.10) and the spaces and bilinear forms of this section satisfies*

$$s(u,u) \leq s(P_{hyl}u,u) \ \leq \ C(1+\log(H/h))^2 s(u,u),$$

where C is independent not only of the mesh size and the number of substructures, but also of the values ρ_i of the coefficient of (4.3).

Fig. 8. Fundamental condition number estimate for BDD (yes, not Schwarz!), taken from [74].

difficult for me to develop an understanding of what these methods really were, why they were all used as preconditioners, and how I should try to use these techniques to develop new DD methods, and maybe also apply them to different, for example time dependent or time harmonic PDEs, especially since those PDEs were often not explicitly mentioned, all was encoded in bilinear forms. The learning curve was incredibly steep and seemed almost insurmountable. I therefore started to look into the earlier literature on DD methods to try to understand where these so highly successful DD preconditioners with incredibly sophisticated mathematical condition number estimates were coming from. What are the pieces, is it possible to separate them and thus get to a simpler understanding of why and how they converge, preferably written at the continous level with a domain decomposition directly for the PDE?

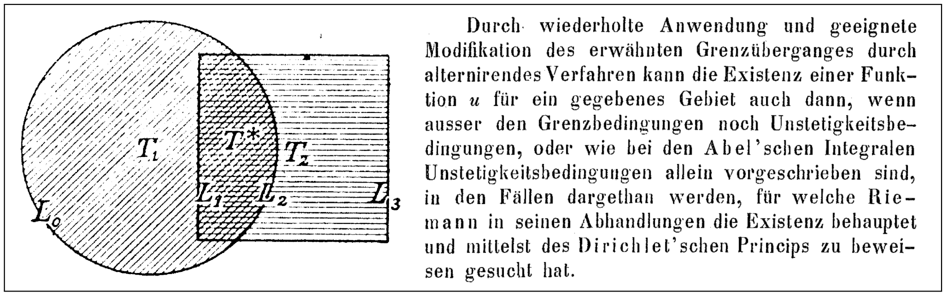
2 Seminal contributions to DD which use only one technique and are thus ideal to start with

> *"In this chapter the reader is encouraged to discover the mathematical foundation of domain decomposition methods, which are based on partitions of the computational domain into subdomains with or without overlap."*
>
> Quarteroni and Valli [69], 1999.

There were indeed also manuscripts which did not introduce domain decomposition methods as preconditioners, but as iterative methods that solved subdomain problems and then combined these subdomain solutions iteratively to get better and better approximations of the solution to the underlying PDE to be solved. These methods were written at the continuous level for a concrete PDE written in strong form, like I had learned in my graduate courses at Stanford. There are also four main classes of such domain decomposition methods which I now briefly introduce with a detailed convergence analysis for a simple model problem.

2.1 The Alternating Schwarz Method (Schwarz 1869)

The first class are the *Schwarz methods*, going back to *Hermann Amandus Schwarz* [70], who invented the method to close a gap in *Riemann's* original



Durch wiederholte Anwendung und geeignete Modifikation des erwähnten Grenzüberganges durch alternirendes Verfahren kann die Existenz einer Funktion *u* für ein gegebenes Gebiet auch dann, wenn ausser den Grenzbedingungen noch Unstetigkeitsbedingungen, oder wie bei den Abel'schen Integralen Unstetigkeitsbedingungen allein vorgeschrieben sind, in den Fällen dargethan werden, für welche Riemann in seinen Abhandlungen die Existenz behauptet und mittelst des Dirichlet'schen Princips zu beweisen gesucht hat.

Fig. 9. Original domain decomposition figure from 1870 for the alternating method of Hermann Amandus Schwarz and his comment: "By repeated application and suitable modification of the mentioned limiting process by the alternating procedure, the existence of a function u for a given domain can be proved even if, in addition to the boundary conditions, discontinuity conditions, or, as in the case of Abel's integrals, discontinuity conditions alone are prescribed, in the cases for which Riemann has asserted the existence in his treatises, and sought to prove it by means of the Dirichlet principle".

proof of the famous *Riemann Mapping Theorem*, see Figure 9, and [52, 53] for more information on the history of this important invention. In order to introduce the alternating Schwarz method, we consider a Poisson problem in 1D,

$$\begin{aligned} \partial_{xx}u &= f \quad \text{in } \Omega := (0,1),\\ u(0) &= g_l,\\ u(1) &= g_r. \end{aligned} \tag{1}$$

Schwarz methods are based on an overlapping decomposition of the domain Ω into subdomains Ω_j, which for Schwarz were a disk and a rectagle as in Figure 9 on the left. For our 1D example (1) we consider the two subdomains $\Omega_1 := (0,\beta)$ and $\Omega_2 := (\alpha,1)$, $\alpha < \beta$, and solve (1) by iteration: given an initial guess $u_2^0(\beta)$, the alternating Schwarz method computes for iteration index $n = 1, 2, \dots$ the sequence of subdomain solutions

$$\begin{aligned} \partial_{xx}u_1^n &= f \quad\quad \text{in } \Omega_1, & \partial_{xx}u_2^n &= f \quad\quad \text{in } \Omega_2,\\ u_1^n(0) &= g_l, & u_2^n(\alpha) &= u_1^n(\alpha),\\ u_1^n(\beta) &= u_2^{n-1}(\beta), & u_2^n(1) &= g_r. \end{aligned} \tag{2}$$

We show in Fig. 10 a graphical illustration of the convergence of the alternating Schwarz method starting with a zero initial guess. We see that the iterates u_j^n seem to converge to the solution u. In order to better understand the convergence, we introduce the error $e_j^n := u - u_j^n$, $j = 1, 2$, $n = 1, 2, \dots$, which satisfies within the subdomains the equation

$$\partial_{xx}e_j^n = \partial_{xx}(u - u_j^n) = f - f = 0 \quad \text{in } \Omega_j,$$

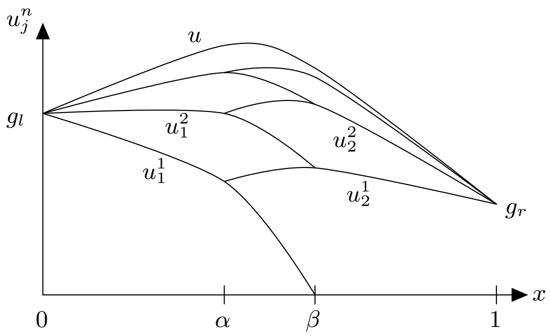


Fig. 10. Example of the alternating Schwarz method in 1D.

i.e. the error satisfies a homogeneous equation, independently of the right hand side function f. Similarly, on the outer boundaries of the domain Ω, we obtain

$$\begin{aligned} e_1^n(0) &= u(0) - u_1^n(0) = g_l - g_l = 0, \\ e_2^n(1) &= u(1) - u_2^n(1) = g_r - g_r = 0, \end{aligned}$$

so the error also does not depend on the boundary values g_l and g_r. On the interfaces at $x = \alpha$ and $x = \beta$, we get

$$\begin{aligned} e_1^n(\beta) &= u(\beta) - u_1^n(\beta) = u(\beta) - u_2^{n-1}(\beta) = e_2^{n-1}(\beta), \\ e_2^n(\alpha) &= u(\alpha) - u_2^n(\alpha) = u(\alpha) - u_1^n(\alpha) = e_1^n(\alpha). \end{aligned}$$

Therefore, the error satisfies the homogeneous alternating Schwarz algorithm

$$\begin{array}{llll} \partial_{xx} e_1^n = 0 & \text{in } \Omega_1, & \partial_{xx} e_2^n = 0 & \text{in } \Omega_2, \\ e_1^n(0) = 0, & & e_2^n(\alpha) = e_1^n(\alpha), & \\ e_1^n(\beta) = e_2^{n-1}(\beta), & & e_2^n(1) = 0. & \end{array} \tag{3}$$

The solutions are simply linear functions, namely

$$e_1^n(x) = \frac{x}{\beta} e_2^{n-1}(\beta), \quad e_2^n(x) = \frac{1-x}{1-\alpha} e_1^n(\alpha),$$

which implies when evaluating the first relation at $x = \alpha$ and inserting the second relation evaluated at $x = \beta$ that

$$e_1^n(\alpha) = \frac{\alpha}{\beta} e_2^{n-1}(\beta) = \frac{\alpha}{\beta} \frac{1-\beta}{1-\alpha} e_1^{n-1}(\alpha),$$

and similarly we also obtain

$$e_2^n(\beta) = \frac{1-\beta}{1-\alpha} e_1^{n-1}(\alpha) = \frac{\alpha}{\beta} \frac{1-\beta}{1-\alpha} e_2^{n-1}(\beta).$$

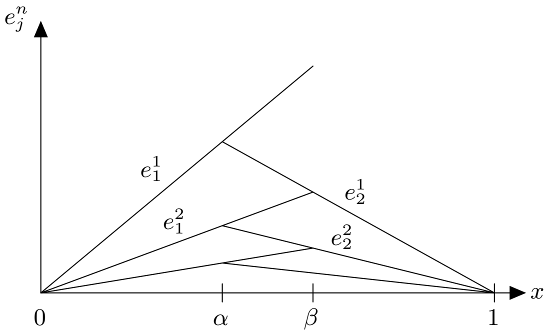


Fig. 11. Error of the alternating Schwarz method in 1D.

We see that the Schwarz method converges if and only if the convergence factor ρ satisfies

$$\rho := \frac{\alpha}{\beta}\frac{1-\beta}{1-\alpha} < 1. \tag{4}$$

This is equivalent to $\alpha < \beta$, i.e. the method needs overlap to work. We show in Fig. 11 how the error in this alternating Schwarz method decreases, which illustrates very well the linear convergence with the same factor ρ after each alternating iteration, and one can also picture how small overlap will make the method slow, and large overlap will make the method fast.

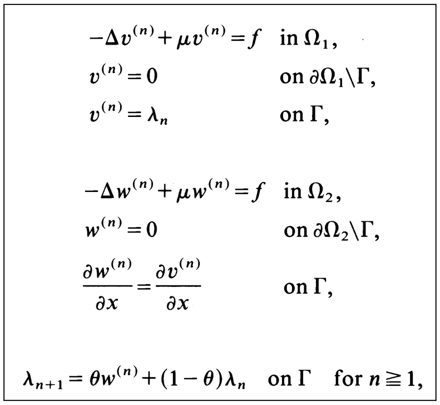
In 1988, Lions introduced a parallel variant of the alternating Schwarz method [63], now called the parallel Schwarz method (of Lions), by replacing the transmission condition for the second subdomain in (2), which is $u_2^n(\alpha) = u_1^n(\alpha)$ in the alternating Schwarz method, by

$$u_2^n(\alpha) = u_1^{n-1}(\alpha). \tag{5}$$

Now the method needs two initial guesses to start, $u_1^0(\alpha)$ and $u_2^0(\beta)$, but the two subdomain solves can be performed in parallel. This is not so interesting for two subdomains, since in this case the parallel method just computes as a subsequence the alternating iterates, and another subsequence of alternating iterates starting on the other subdomain. For many subdomains however, the parallel Schwarz method is a very interesting way to use many processors, one per subdomain, simultaneously to solve a large scale problem.

2.2 The Dirichlet-Neumann Method

The Schwarz methods we have seen in the previous subsection need overlap for convergence. In order to obtain a DD method that works without overlap, one can not only use Dirichlet transmission conditions. More than a century after Schwarz, we find in [32] the so called Dirichlet-Neumann method, which is based on a non-overlapping decomposition, see Figure 12, and also [76], where



$$\begin{aligned}
&-\Delta v^{(n)}+\mu v^{(n)}=f && \text{in } \Omega_1,\\
&v^{(n)}=0 && \text{on } \partial\Omega_1\backslash\Gamma,\\
&v^{(n)}=\lambda_n && \text{on } \Gamma,\\[1em]
&-\Delta w^{(n)}+\mu w^{(n)}=f && \text{in } \Omega_2,\\
&w^{(n)}=0 && \text{on } \partial\Omega_2\backslash\Gamma,\\
&\frac{\partial w^{(n)}}{\partial x}=\frac{\partial v^{(n)}}{\partial x} && \text{on } \Gamma,\\[1em]
&\lambda_{n+1}=\theta w^{(n)}+(1-\theta)\lambda_n \quad \text{on } \Gamma \quad \text{for } n\geqq 1,
\end{aligned}$$

Fig. 12. The formulation of the Dirichlet-Neumann method from [32].

it is presented as an alternative to the alternating Schwarz methods without overlap. For our 1D Poisson model problem (1), if we consider the two non-overlapping subdomains $\Omega_1 := (0,\alpha)$ and $\Omega_2 := (\alpha,1)$, the Dirichlet-Neumann method would start with an initial guess $u_2^0(\alpha)$, and then compute for iteration index $n = 1, 2, \ldots$ in an alternating fashion like the alternating Schwarz method a Dirichlet but then a Neumann problem,

$$\begin{array}{rll@{\qquad}rll}
\partial_{xx}u_1^n &= f & \text{in } \Omega_1, & \partial_{xx}u_2^n &= f & \text{in } \Omega_2,\\
u_1^n(0) &= g_l, & & -\partial_x u_2^n(\alpha) &= -\partial_x u_1^n(\alpha), &\\
u_1^n(\alpha) &= u_2^{n-1}(\alpha), & & u_2^n(1) &= g_r. &
\end{array} \tag{6}$$

We wrote here $-\partial_x$ instead of ∂_x which would have been the same algorithm, in order to emphasize that in more general situations, it is the unit outward normal derivative which is used on the Neumann interface, and for the second subdomain at α the unit outward normal derivative at the left border $x = \alpha$ is $-\partial_x$. As for the alternating Schwarz method, we can write the error equations for the Dirichlet-Neumann method,

$$\begin{array}{rll@{\qquad}rll}
\partial_{xx}e_1^n &= 0 & \text{in } \Omega_1, & \partial_{xx}e_2^n &= 0 & \text{in } \Omega_2,\\
e_1^n(0) &= 0, & & -\partial_x e_2^n(\alpha) &= -\partial_x e_1^n(\alpha), &\\
e_1^n(\alpha) &= e_2^{n-1}(\alpha), & & e_2^n(1) &= 0, &
\end{array} \tag{7}$$

and we obtain for the errors the solutions

$$e_1^n(x) = \frac{x}{\alpha}e_2^{n-1}(\alpha), \quad e_2^n(x) = -(1-x)\partial_x e_1^n(\alpha),$$

as one can easily check by inspection. This implies that at $x = \alpha$

$$e_1^n(\alpha) = e_2^{n-1}(\alpha) = -(1-\alpha)\partial_x e_1^{n-1}(\alpha) = -\frac{1-\alpha}{\alpha}e_2^{n-2}(\alpha) = \frac{\alpha-1}{\alpha}e_1^{n-1}(\alpha),$$

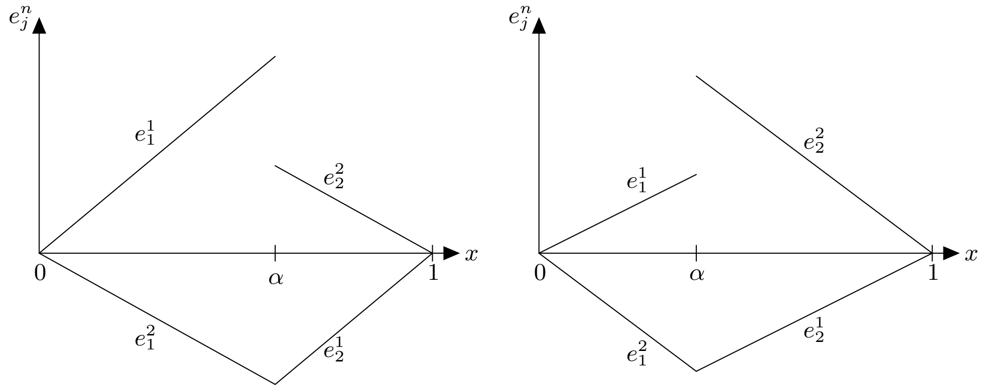


Fig. 13. Error of the Dirichlet-Neumann method in 1D. Left with $\alpha > \frac{1}{2}$ where it converges, and right with $\alpha < \frac{1}{2}$ where it does not.

where we explicitly computed the derivative in the second last step, and used in the last step that $e_1^{n-1}(\alpha) = e_1^{n-2}(\alpha)$. Similarly we also obtain

$$e_2^n(\alpha) = -(1-\alpha)\partial_x e_1^n(\alpha) = \frac{\alpha-1}{\alpha} e_2^{n-1}(\alpha).$$

We see that the Dirichlet-Neumann method converges if and only if the convergence factor ρ satisfies

$$\rho := \left|\frac{\alpha-1}{\alpha}\right| = \frac{1-\alpha}{\alpha} < 1 \quad \text{for } \alpha \in \Omega = (0,1).$$

So if $\alpha > \frac{1}{2}$, the Dirichlet-Neumann method converges, whereas for $\alpha \leq \frac{1}{2}$ it does not, as illustrated in Fig. 13. The Dirichlet-Neumann method therefore can not just be used as is, and one adds a relaxation parameter θ, as it was already shown in Figure 12: one replaces the transmission condition of the Dirichlet subdomain in (6) by

$$u_1^n(\alpha) = g^{n-1},$$

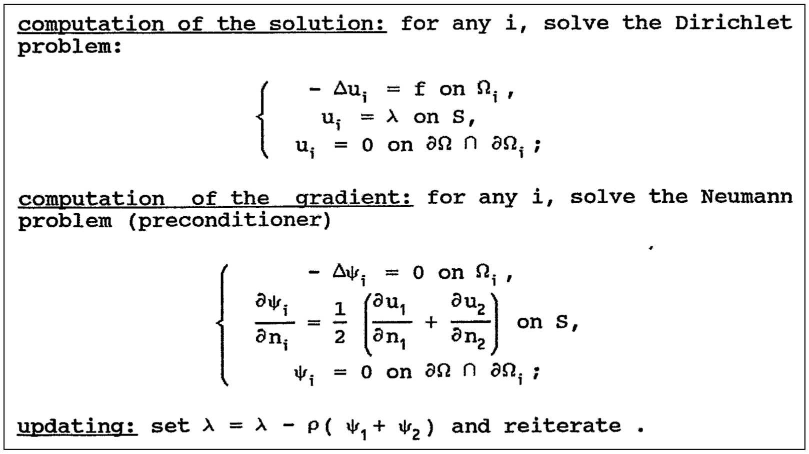
and thus needs to start the algorithm with an initial guess g^0, and then computes the new g^n by a linear combination of the old one and the newly computed value on the second subdomain,

$$g^n = \theta g^{n-1} + (1-\theta) u_2^n(\alpha).$$

The convergence factor of the Dirichlet-Neumann method with relaxation becomes

$$\rho = \theta + (1-\theta)\frac{\alpha-1}{\alpha},$$

just a weighed average of 1 and the old convergence factor, and now one can choose θ to make the method converge for all α. The best choice θ^* makes the



computation of the solution: for any i, solve the Dirichlet problem:

$$\begin{cases} -\Delta u_i = f \text{ on } \Omega_i\,, \\ u_i = \lambda \text{ on } S, \\ u_i = 0 \text{ on } \partial\Omega \cap \partial\Omega_i\,; \end{cases}$$

computation of the gradient: for any i, solve the Neumann problem (preconditioner)

$$\begin{cases} -\Delta\psi_i = 0 \text{ on } \Omega_i\,, \\ \dfrac{\partial\psi_i}{\partial n_i} = \dfrac{1}{2}\left(\dfrac{\partial u_1}{\partial n_1} + \dfrac{\partial u_2}{\partial n_2}\right) \text{ on } S, \\ \psi_i = 0 \text{ on } \partial\Omega \cap \partial\Omega_i\,; \end{cases}$$

updating: set $\lambda = \lambda - \rho(\,\psi_1 + \psi_2)$ and reiterate .

Fig. 14. The invention of the Neumann-Neumann method from [6].

convergence factor as small as possible, and a small miracle happens in 1D, since we can make the convergence factor vanish identically: $\rho \equiv 0$ is obtained for

$$\theta^* + (1-\theta^*)\frac{\alpha-1}{\alpha} = 0 \quad\Longleftrightarrow\quad \theta^* = 1-\alpha.$$

With this optimal choice θ^* of the relaxation parameter, the Dirichlet-Neumann method converges in two iterations, it becomes a direct solver, something which is not possible with the classical alternating Schwarz method[2].

2.3 The Neumann-Neumann Method

The third class of DD methods are the Neumann-Neumann methods invented by Bourgat, Glowinski, Le Tallec and Vidrascu [6], see Figure 14. These methods also use non-overlapping subdomains, but solve two problems on each subdomain, first Dirichlet problems starting with an initial guess g^0 on the interface, for our 1D model problem

$$\begin{array}{llll} \partial_{xx}u_1^n = f & \text{in } \Omega_1, & \partial_{xx}u_2^n = f & \text{in } \Omega_2, \\ u_1^n(0) = g_l, & & u_2^n(\alpha) = g^{n-1}, & \\ u_1^n(\alpha) = g^{n-1}, & & u_2^n(1) = g_r, & \end{array} \tag{8}$$

followed by Neumann correction problems,

$$\begin{array}{llll} \partial_{xx}\psi_1^n = 0 & \text{in } \Omega_1, & \partial_{xx}\psi_2^n = 0 & \text{in } \Omega_2, \\ \psi_1^n(0) = 0, & & -\partial_x\psi_2^n(\alpha) = \partial_x u_1^n(\alpha) - \partial_x u_2^n(\alpha), & \\ \partial_x\psi_1^n(\alpha) = \partial_x u_1^n(\alpha) - \partial_x u_2^n(\alpha), & & \psi_2^n(1) = 0, & \end{array} \tag{9}$$

[2] It is this miracle which in higher spatial dimensions still happens for high frequencies and $\theta = \frac{1}{2}$ and makes this into a very powerful solver that converges independently of the mesh size, see [22, Section 4.7]. Also when using Krylov acceleration, optimizing θ does not contribute anything, see [22, end of Section 4.7].

and then also a relaxation step is needed, since otherwise convergence can not be ensured,

$$g^n = g^{n-1} - \theta(\psi_1^n(\alpha) + \psi_2^n(\alpha)).$$

The error equations for the Neumann-Neumann methods, denoting with $d^n := g - g^n$ for $g := u(\alpha)$ the error at the interface, are

$$\begin{array}{rlrl} \partial_{xx} e_1^n = 0 & \text{in } \Omega_1, & \partial_{xx} e_2^n = 0 & \text{in } \Omega_2, \\ e_1^n(0) = 0, & & e_2^n(\alpha) = d^{n-1}, & \\ e_1^n(\alpha) = d^{n-1}, & & e_2^n(1) = 0, & \end{array} \tag{10}$$

followed by the Neumann correction problems

$$\begin{array}{rlrl} \partial_{xx} \psi_1^n = 0 & \text{in } \Omega_1, & \partial_{xx} \psi_2^n = 0 & \text{in } \Omega_2, \\ \psi_1^n(0) = 0, & & -\partial_x \psi_2^n(\alpha) = \partial_x e_1^n(\alpha) - \partial_x e_2^n(\alpha), & \\ \partial_x \psi_1^n(\alpha) = \partial_x e_1^n(\alpha) - \partial_x e_2^n(\alpha), & & \psi_2^n(1) = 0, & \end{array} \tag{11}$$

and the relaxation step for the error,

$$d^n = d^{n-1} - \theta(\psi_1^n(\alpha) + \psi_2^n(\alpha)).$$

As for the Dirichlet-Neumann method, we can easily compute the linear solutions for the error,

$$e_1^n(x) = \frac{x}{\alpha} d^{n-1}, \quad e_2^n(x) = \frac{1-x}{1-\alpha} d^{n-1},$$

which gives for the Neumann conditions in the correction problems

$$\partial_x e_1^n(\alpha) - \partial_x e_2^n(\alpha) = \frac{1}{\alpha} d^{n-1} + \frac{1}{1-\alpha} d^{n-1} = \frac{1}{\alpha(1-\alpha)} d^{n-1}.$$

The corrections therefore become

$$\psi_1^n(x) = \frac{x}{\alpha(1-\alpha)} d^{n-1}, \quad \psi_2^n(x) = \frac{1-x}{\alpha(1-\alpha)} d^{n-1},$$

which leads to the relaxation step

$$d^n = d^{n-1} - \theta(\frac{\alpha}{\alpha(1-\alpha)} + \frac{1-\alpha}{\alpha(1-\alpha)}) d^{n-1} = (1 - \theta \frac{1}{\alpha(1-\alpha)}) d^{n-1}.$$

Hence, the convergence factor ρ of the Neumann-Neumann method is

$$\rho = 1 - \theta \frac{1}{\alpha(1-\alpha)}, \tag{12}$$

and like for the Dirichlet-Neumann method, we can obtain convergence in 2 steps[3] if we choose θ such that $\rho \equiv 0$, i.e.

$$\theta = \theta^* = \alpha(1-\alpha).$$

Note that BDD from Subsection 1.4 is in fact a Neumann-Neumann method, with a simple piecewise coarse space added.

[3] It is again this property that makes the Neumann-Neumann method into such a powerful solver that converges independently of the mesh size, like the Dirichlet-Neumann method, see [22, Section 4.8].

2.4 The FETI Method

The last class of DD methods are the FETI (Finite Element Tearing and Interconnect) methods [31], which have been invented in the context of writing the PDE as a minimization problem. FETI methods turn out however to correspond to simply inverting the Dirichlet and Neumann solves in the Neumann-Neumann method. In our two subdomain example, they would start with Neumann solves on the subdomains (while the Neumann-Neumann method starts with Dirichlet solves, which does not seem natural but historically the method was named that way), using as initial guess g^0 on the interface,

$$\begin{array}{rlrl} \partial_{xx}u_1^n = f & \text{in } \Omega_1, & \partial_{xx}u_2^n = f & \text{in } \Omega_2, \\ u_1^n(0) = g_l, & & \partial_x u_2^n(\alpha) = g^{n-1}, & \\ \partial_x u_1^n(\alpha) = g^{n-1}, & & u_2^n(1) = g_r, & \end{array} \tag{13}$$

followed by two Dirichlet correction problems,

$$\begin{array}{rlrl} \partial_{xx}\psi_1^n = 0 & \text{in } \Omega_1, & \partial_{xx}\psi_2^n = 0 & \text{in } \Omega_2, \\ \psi_1^n(0) = 0, & & \psi_2^n(\alpha) = u_1^n(\alpha) - u_2^n(\alpha), & \\ \psi_1^n(\alpha) = u_1^n(\alpha) - u_2^n(\alpha), & & \psi_2^n(1) = 0, & \end{array} \tag{14}$$

and then again a relaxation step is needed,

$$g^n = g^{n-1} - \theta(\partial_x\psi_1^n(\alpha) - \partial_x\psi_2^n(\alpha)).$$

The error equations for FETI, again denoting with $d^n := g - g^n$ for $g := \partial_x u(\alpha)$ the error at the interface, are

$$\begin{array}{rlrl} \partial_{xx}e_1^n = 0 & \text{in } \Omega_1, & \partial_{xx}e_2^n = 0 & \text{in } \Omega_2, \\ e_1^n(0) = 0, & & \partial_x e_2^n(\alpha) = d^{n-1}, & \\ \partial_x e_1^n(\alpha) = d^{n-1}, & & e_2^n(1) = 0, & \end{array} \tag{15}$$

followed by the Dirichlet correction problems

$$\begin{array}{rlrl} \partial_{xx}\psi_1^n = 0 & \text{in } \Omega_1, & \partial_{xx}\psi_2^n = 0 & \text{in } \Omega_2, \\ \psi_1^n(0) = 0, & & \psi_2^n(\alpha) = e_1^n(\alpha) - e_2^n(\alpha), & \\ \psi_1^n(\alpha) = e_1^n(\alpha) - e_2^n(\alpha), & & \psi_2^n(1) = 0, & \end{array} \tag{16}$$

and the relaxation step for the error,

$$d^n = d^{n-1} - \theta(\partial_x\psi_1^n(\alpha) - \partial_x\psi_2^n(\alpha)).$$

As for the Neumann-Neumann method, we can easily compute the linear solutions for the error,

$$e_1^n(x) = xd^{n-1}, \quad e_2^n(x) = -(1-x)d^{n-1},$$

which gives for the Dirichlet conditions in the correction problems

$$e_1^n(\alpha) - e_2^n(\alpha) = (\alpha + (1-\alpha))d^{n-1} = d^{n-1}.$$

The corrections therefore become

$$\psi_1^n(x) = \frac{x}{\alpha}d^{n-1}, \quad \psi_2^n(x) = \frac{1-x}{1-\alpha}d^{n-1},$$

which leads to the relaxation step

$$d^n = d^{n-1} - \theta(\frac{1}{\alpha} + \frac{1}{1-\alpha})d^{n-1} = (1 - \theta\frac{1}{\alpha(1-\alpha)})d^{n-1}.$$

We see that FETI has the same convergence factor ρ as the Neumann-Neumann method,

$$\rho = 1 - \theta\frac{1}{\alpha(1-\alpha)}, \tag{17}$$

and like for the Neumann-Neumann method, FETI converges in 2 steps[4] if we choose

$$\theta = \theta^* = \alpha(1-\alpha).$$

3 DD solvers are obtained by discretizing these

> *"In numerical computation one will, of course, always be working with some finite dimensional discretization of the PDE."*
>
> Smith, Bjørstad, Gropp [71], 1996.

To use Schwarz, Dirichlet-Neumann, Neumann-Neumann and FETI methods on the computer, one has to discretize them. For example for the 1D model Poisson problem of the Schwarz method we have seen in (2), if we discretize the problem with a finite difference method and 6 interior grid points, i.e. mesh size $h := \frac{1}{7}$, we obtain the linear system

$$\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix}\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{pmatrix} = \begin{pmatrix} f_1 - \frac{1}{h^2}g_l \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 - \frac{1}{h^2}g_r \end{pmatrix}. \tag{18}$$

If we choose for the two subdomains in the alternating Schwarz method (2) as subdomain boundaries $\alpha := 2h$ and $\beta = 5h$, as indicated in Fig. 15, we obtain

[4] Again it is this property that makes FETI into such a powerful solver, and its relation with Neumann-Neumann explains why the condition number estimate in Fig. 6 for FETI and in Fig. 8 for BDD are the same.

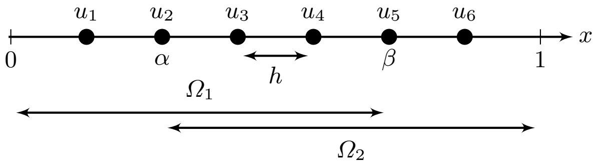


Fig. 15. Discretization of the alternating Schwarz method with the two subdomains $\Omega_1 = (0, \beta)$ and $\Omega_2 = (\alpha, 1)$.

from (2) the discretized alternating Schwarz method

$$\begin{aligned}
\underbrace{\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & 1 & -2 & 1 \\ & & 1 & -2 \end{bmatrix}}_{A_1}
\underbrace{\begin{pmatrix} u_{1,1}^n \\ u_{1,2}^n \\ u_{1,3}^n \\ u_{1,4}^n \end{pmatrix}}_{\boldsymbol{u}_1^n}
&= \underbrace{\begin{pmatrix} f_1 - \frac{1}{h^2} g_l \\ f_2 \\ f_3 \\ f_4 - \frac{1}{h^2} u_{2,5}^{n-1} \end{pmatrix}}_{\boldsymbol{f}_1 + \frac{1}{h^2} u_{2,5}^{n-1} \boldsymbol{e}_4}, \\
\underbrace{\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & 1 & -2 & 1 \\ & & 1 & -2 \end{bmatrix}}_{A_2}
\underbrace{\begin{pmatrix} u_{2,3}^n \\ u_{2,4}^n \\ u_{2,5}^n \\ u_{2,6}^n \end{pmatrix}}_{\boldsymbol{u}_2^n}
&= \underbrace{\begin{pmatrix} f_3 - \frac{1}{h^2} u_{1,2}^n \\ f_4 \\ f_5 \\ f_6 - \frac{1}{h^2} g_r \end{pmatrix}}_{\boldsymbol{f}_2 + \frac{1}{h^2} u_{1,2}^n \boldsymbol{e}_1}.
\end{aligned} \tag{19}$$

This method can now be interpreted at the linear algebra level, and to do so, it is useful to write the discretized linear system (18) in augmented form, by writing the equations in the overlap twice,

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & & & & \\ 1 & -2 & 1 & & & & & \\ & 1 & -2 & 1 & & & & \\ & & 1 & -2 & & & 1 & \\ & 1 & & & -2 & 1 & & \\ & & & & 1 & -2 & 1 & \\ & & & & & 1 & -2 & 1 \\ & & & & & & 1 & -2 \end{bmatrix}}_{A}
\underbrace{\begin{pmatrix} u_{1,1} \\ u_{1,2} \\ u_{1,3} \\ u_{1,4} \\ u_{2,3} \\ u_{2,4} \\ u_{2,5} \\ u_{2,6} \end{pmatrix}}_{\boldsymbol{u}}
= \underbrace{\begin{pmatrix} f_1 + \frac{1}{h^2} g_l \\ f_2 \\ f_3 \\ f_4 \\ f_3 \\ f_4 \\ f_5 \\ f_6 + \frac{1}{h^2} g_r \end{pmatrix}}_{\boldsymbol{f}}. \tag{20}$$

If we apply a stationary iteration to this augmented system with matrix splitting $A = M - N$,

$$M\boldsymbol{u}^n = N\boldsymbol{u}^{n-1} + \boldsymbol{f}, \tag{21}$$

and use as splitting the block Gauss-Seidel splitting,

$$M := \begin{bmatrix} A_1 & 0 \\ A_{21} & A_2 \end{bmatrix}, \quad N := \begin{bmatrix} 0 & -A_{12} \\ 0 & 0 \end{bmatrix}, \tag{22}$$

where the matrix blocks A_1 and A_2 correspond to the Schwarz subdomain matrices in (19), and A_{12} and A_{21} are the remaining off diagonal blocks in the augmented system (20), the resulting block Gauss-Seidel method coincides with the discretized alternating Schwarz method (19). Similarly, the discretized parallel Schwarz method is obtained when using a block Jacobi method for solving the augmented system, i.e. one uses the matrix splitting iteration (21) with the block Jacobi splitting,

$$M := \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, \quad N := \begin{bmatrix} 0 & -A_{12} \\ -A_{21} & 0 \end{bmatrix}, \tag{23}$$

we obtain a discretization of the parallel Schwarz method with the second transmission condition changed into (5).

Remark 1. Note that the overlap of these methods is $\beta - \alpha = 3h$, as one can see in Fig. 15. This overlap corresponds at the algebraic level to 2 nodes overlap. With minimal overlap, i.e. zero nodes and $\alpha = 3h$ and $\beta = 4h$ this gives still as overlap $\beta - \alpha = h$. Note that with minimal overlap, the augmented system (20) is identical to the original system (18).

As we did at the continuous level, we can also introduce the error $\boldsymbol{e}^n := \boldsymbol{u} - \boldsymbol{u}^n$ at the discrete level, and since the solution $\boldsymbol{u}$ satisfies the same equation (21) as the iterates $\boldsymbol{u}^n$, we obtain by taking the difference the discrete error equations

$$\boldsymbol{e}^n = M^{-1} N \boldsymbol{e}^{n-1}, \tag{24}$$

which do not depend on the source terms $\boldsymbol{f}$, like at the continuous level. Convergence is therefore fast if the spectral radius (largest eigenvalue in modulus) of the iteration matrix, $\rho(M^{-1}N)$ is small. Using that $A = M - N$, we get

$$M^{-1}N = M^{-1}(M - A) = I - M^{-1}A,$$

and thus for fast convergence we need

$$\rho(I - M^{-1}A) \ll 1. \tag{25}$$

Getting back to the DD methods, not only the alternating and parallel Schwarz methods, as we have seen, can be written at the linear algebra level as a matrix splitting $A = M - N$, also the Dirichlet-Neumann, Neumann-Neumann and FETI methods can be written like this, always using an appropriate augmented system formulation like in (20) for the alternating Schwarz method.

It is however also possible to write the discretized alternating Schwarz method without introducing an augmented system. If we introduce for our discretized 1D Poisson example the rectangular restriction matrices

$$R_1 := \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \end{bmatrix} \quad \text{and} \quad R_2 := \begin{bmatrix} & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix},$$

and consider the original, not augmented system (18), i.e.

$$\underbrace{\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix}}_{A}\underbrace{\begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{pmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{pmatrix} f_1 - \frac{1}{h^2}g_l \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 - \frac{1}{h^2}g_r \end{pmatrix}}_{\boldsymbol{f}}, \tag{26}$$

then the discrete iteration

$$\begin{aligned} \boldsymbol{u}^{n-\frac{1}{2}} &= \boldsymbol{u}^{n-1} + R_1^T A_1^{-1} R_1(\boldsymbol{f} - A\boldsymbol{u}^{n-1}), \\ \boldsymbol{u}^{n} &= \boldsymbol{u}^{n-\frac{1}{2}} + R_2^T A_2^{-1} R_2(\boldsymbol{f} - A\boldsymbol{u}^{n-\frac{1}{2}}), \end{aligned} \tag{27}$$

where the matrices $A_1 := R_1 A R_1^T$ and $A_2 := R_2 A R_2^T$ are algebraically defined, is called the Multiplicative Schwarz method, and it can be used with any linear system $A\boldsymbol{u} = \boldsymbol{f}$. This name comes from looking at the error $\boldsymbol{e}^n := \boldsymbol{u} - \boldsymbol{u}^n$ which satisfies

$$\begin{aligned} \boldsymbol{e}^{n-\frac{1}{2}} &= \boldsymbol{e}^{n-1} - R_1^T A_1^{-1} R_1 A\boldsymbol{e}^{n-1} =: (I - P_1)\boldsymbol{e}^{n-1}, \\ \boldsymbol{e}^{n} &= \boldsymbol{e}^{n-\frac{1}{2}} - R_2^T A_2^{-1} R_2 A\boldsymbol{e}^{n-\frac{1}{2}} =: (I - P_2)\boldsymbol{e}^{n-\frac{1}{2}}, \end{aligned} \tag{28}$$

and inserting the first equation into the second on the right we get

$$\boldsymbol{e}^n = (I - P_2)(I - P_1)\boldsymbol{e}^{n-1}, \tag{29}$$

i.e. the complementary projection operations $(I - P_j)$ are applied in multiplied form, hence 'Multiplicative' in the Multiplicative Schwarz method. But why is there the name 'Schwarz' in the method? This is because the iteration (27) leads to the same iterates as the discretization of the alternating Schwarz method (19), as one can see as follows for our 1D example: for the residual in the $n - \frac{1}{2}$ step, we get

$$\boldsymbol{f} - A\boldsymbol{u}^{n-1} = \begin{bmatrix} f_1 - \frac{1}{h^2}g_l \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 - \frac{1}{h^2}g_r \end{bmatrix} - \begin{bmatrix} A_1 \begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ u_4^{n-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{h^2}u_5^{n-1} \end{bmatrix} \\ \begin{bmatrix} \frac{1}{h^2}u_4^{n-1} \\ 0 \end{bmatrix} + B \begin{bmatrix} u_5^{n-1} \\ u_6^{n-1} \end{bmatrix} \end{bmatrix},$$

where B is the remaining part of the matrix A and of no importance, since the following restriction step removes it,

$$R_1(\boldsymbol{f} - A\boldsymbol{u}^{n-1}) = \begin{bmatrix} f_1 - \frac{1}{h^2}g_l \\ f_2 \\ f_3 \\ f_{b-1} - \frac{1}{h^2}u_5^{n-1} \end{bmatrix} - A_1 \begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ u_4^{n-1} \end{bmatrix}.$$

Next the subdomain solve A_1^{-1} is applied which cancels with the matrix A_1 in the second term,

$$A_1^{-1}R_1(\boldsymbol{f}-A\boldsymbol{u}^{n-1})=\underbrace{A_1^{-1}\begin{bmatrix} f_1-\frac{1}{h^2}g_l \\ f_2 \\ f_3 \\ f_4-\frac{1}{h^2}u_5^{n-1}\end{bmatrix}}_{\boldsymbol{u}_1^n}-\begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ u_4^{n-1}\end{bmatrix}=\begin{bmatrix} u_{1,1}^n \\ u_{1,2}^n \\ u_{1,3}^n \\ u_{1,4}^n\end{bmatrix}-\begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ u_4^{n-1}\end{bmatrix}, \qquad (30)$$

where we see that precisely the subdomain solve $\boldsymbol{u}_1^n$ of the discretized alternating Schwarz method (19) appeared with transmission condition u_5^{n-1}! Continuing with the Multiplicative Schwarz algorithm, we extend (30) with R_1^T by patching zeros and add it to $\boldsymbol{u}^{n-1}$ to get another interesting cancellation of the old iterate,

$$\boldsymbol{u}^{n-\frac{1}{2}}=\begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ u_4^{n-1} \\ u_5^{n-1} \\ u_6^{n-1}\end{bmatrix}+\begin{bmatrix} \begin{bmatrix} u_{1,1}^n \\ u_{1,2}^n \\ u_{1,3}^n \\ u_{1,1}^n\end{bmatrix}-\begin{bmatrix} u_1^{n-1} \\ u_2^{n-1} \\ u_3^{n-1} \\ u_4^{n-1}\end{bmatrix} \\ 0 \\ 0\end{bmatrix}=\begin{bmatrix} u_{1,1}^n \\ u_{1,2}^n \\ u_{1,3}^n \\ u_{1,4}^n \\ u_5^{n-1} \\ u_6^{n-1}\end{bmatrix}.$$

This shows that the residual computation in the multiplicative Schwarz formulation does two things: first it extracts the Dirichlet transmission condition so the subdomain solve appears, and second it produces a copy of the old iterate so it cancels with the old iterate on the first subdomain which is in the first vector to the right of the equal sign, and can be replaced by the new iterate. Hence in the new global iterate $\boldsymbol{u}^{n-\frac{1}{2}}$, no matter what the content of the entries $1\ldots4$ was, we find the solution of a subdomain solve with boundary condition u_5^{n-1}. For the second part of the algorithm, we find similarly

$$A_2^{-1}R_2(\boldsymbol{f}-A\boldsymbol{u}^{n-\frac{1}{2}})=\underbrace{A_2^{-1}\begin{bmatrix} f_3-\frac{1}{h^2}u_{1,2}^n \\ f_4 \\ f_5 \\ f_6-\frac{1}{h^2}g_l\end{bmatrix}}_{\boldsymbol{u}_2^n}-\begin{bmatrix} u_{1,3}^n \\ u_{1,4}^n \\ u_5^{n-1} \\ u_6^{n-1}\end{bmatrix}=\begin{bmatrix} u_{2,3}^n \\ u_{2,4}^n \\ u_{2,5}^n \\ u_{2,6}^n\end{bmatrix}-\begin{bmatrix} u_{1,3}^n \\ u_{1,4}^n \\ u_5^{n-1} \\ u_6^{n-1}\end{bmatrix},$$

where we see now the second subdomain solve $\boldsymbol{u}_2^n$ appear with boundary condition $u_{1,2}^n$ from the discretized alternating Schwarz method (19). Extending this result with R_2^T by patching zeros and adding it to $\boldsymbol{u}^{n-\frac{1}{2}}$ cancels again the old entries and replaces them with the new subdomain solve on subdomain two,

$$\boldsymbol{u}^{n-1}=\begin{bmatrix} u_{1,1}^n \\ u_{1,2}^n \\ u_{1,3}^n \\ u_{1,4}^n \\ u_5^{n-1} \\ u_6^{n-1}\end{bmatrix}+\begin{bmatrix} 0 \\ 0 \\ \begin{bmatrix} u_{2,3}^n \\ u_{2,4}^n \\ u_{2,5}^n \\ u_{2,6}^n\end{bmatrix}-\begin{bmatrix} u_{1,3}^n \\ u_{2,4}^n \\ u_5^{n-1} \\ u_6^{n-1}\end{bmatrix}\end{bmatrix}=\begin{bmatrix} u_{1,1}^n \\ u_{1,2}^n \\ u_{2,3}^n \\ u_{2,4}^n \\ u_{2,5}^n \\ u_{2,6}^n\end{bmatrix}.$$

Hence the Multiplicative Schwarz method described algebraically in (27) is equivalent to the discretized alternating Schwarz method (19) if the matrix A is a discretization of the partial differential equation and the restriction matrices R_j correspond to the subdomains. This result holds not just for our simple example, but in full generality, see [34, Theorem 3.3].

The Multiplicative Schwarz formulation (27) is very convenient for implementation, but one has to perform more computations than in the mathematically equivalent discretized alternating Schwarz implementation (19): in addition to the subdomain solves, one also needs to compute the residual $\boldsymbol{f} - A\boldsymbol{u}^{n-1}$. Furthermore, in a parallel environment, in Multiplicative Schwarz (27), one has to communicate all variables in the overlap, whereas in the discretized alternating Schwarz implementation (19), one has to communicate only the interface variables. In a production code, one should therefore never use the Multiplicative Schwarz formulation (27), but always implement the discretized alternating Schwarz method (19). The Multiplicative Schwarz formulation (27) is just to be used for theoretical purposes.

We can now also understand the first step in the seminal invention of the Additive Schwarz method (AS) in Fig. 1, since in (29) we have now seen what the product $(I-P_1)(I-P_2)$ means, for a discretized partial differential equation and two subdomains. So how would AS then look like that corresponds to the mysterious equation (3) in Fig. 1? In contrast to the Multiplicative Schwarz method (27), we would have to add the corrections for AS,

$$\boldsymbol{u}^n = \boldsymbol{u}^{n-1} + (R_1^T A_1^{-1} R_1 + R_2^T A_2^{-1} R_2)(\boldsymbol{f} - A\boldsymbol{u}^{n-1}), \tag{31}$$

and the natural question arises if this is then a discretization of the parallel Schwarz method of Lions, which for our example is

$$\begin{aligned}
\underbrace{\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & 1 & -2 & 1 \\ & & 1 & -2 \end{bmatrix}}_{A_1}
\underbrace{\begin{pmatrix} u_{1,1}^n \\ u_{1,2}^n \\ u_{1,3}^n \\ u_{1,4}^n \end{pmatrix}}_{\boldsymbol{u}_1^n}
&= \underbrace{\begin{pmatrix} f_1 - \frac{1}{h^2} g_l \\ f_2 \\ f_3 \\ f_4 - \frac{1}{h^2} u_{2,5}^{n-1} \end{pmatrix}}_{\boldsymbol{f}_1 + \frac{1}{h^2} u_{2,5}^{n-1} \boldsymbol{e}_4}, \\
\underbrace{\frac{1}{h^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & 1 & -2 & 1 \\ & & 1 & -2 \end{bmatrix}}_{A_2}
\underbrace{\begin{pmatrix} u_{2,3}^n \\ u_{2,4}^n \\ u_{2,5}^n \\ u_{2,6}^n \end{pmatrix}}_{\boldsymbol{u}_2^n}
&= \underbrace{\begin{pmatrix} f_3 - \frac{1}{h^2} u_{1,2}^{n-1} \\ f_4 \\ f_5 \\ f_6 - \frac{1}{h^2} g_r \end{pmatrix}}_{\boldsymbol{f}_2 + \frac{1}{h^2} u_{1,2}^{n} \boldsymbol{e}_1}.
\end{aligned} \tag{32}$$

Note that the only difference compared to the discretized alternating Schwarz method (19) is the transmitted data to subdomain Ω_2 which is now $u_{1,2}^{n-1}$ instead of $u_{1,2}^n$ in order for the subdomain solves to be done in parallel. Tracing each step of AS (31) as we did for the Multiplicative Schwarz method, we find

$$\boldsymbol{u}^n=\begin{bmatrix}u_1^{n-1}\\u_2^{n-1}\\u_3^{n-1}\\u_4^{n-1}\\u_5^{n-1}\\u_6^{n-1}\end{bmatrix}+\begin{bmatrix}A_1^{-1}\begin{bmatrix}f_1-\frac{1}{h^2}g_l\\f_2\\f_3\\f_4-\frac{1}{h^2}u_5^{n-1}\end{bmatrix}-\begin{bmatrix}u_1^{n-1}\\u_2^{n-1}\\u_3^{n-1}\\u_4^{n-1}\end{bmatrix}\\0\\0\end{bmatrix}+\begin{bmatrix}0\\0\\A_2^{-1}\begin{bmatrix}f_3-\frac{1}{h^2}u_2^{n-1}\\f_4\\f_5\\f_6-\frac{1}{h^2}g_r\end{bmatrix}-\begin{bmatrix}u_3^{n-1}\\u_4^{n-1}\\u_5^{n-1}\\u_6^{n-1}\end{bmatrix}\end{bmatrix}, \tag{33}$$

or with the subdomain solutions $\boldsymbol{u}_1^n$ and $\boldsymbol{u}_2^n$ from the parallel Schwarz method of Lions (32)

$$\boldsymbol{u}^n=\begin{bmatrix}u_{1,1}^n\\u_{1,2}^n\\u_{1,3}^n\\u_{1,4}^n\\0\\0\end{bmatrix}+\begin{bmatrix}0\\0\\u_{2,3}^n\\u_{2,4}^n\\u_{2,5}^n\\u_{2,6}^n\end{bmatrix}-\begin{bmatrix}0\\0\\u_3^{n-1}\\u_4^{n-1}\\0\\0\end{bmatrix}. \tag{34}$$

So now the magic cancellation we have seen in Multiplicative Schwarz does not work any more in the overlap, the values of the old iterate remain, and the two new subdomain approximations are added, except if the overlap is minimal, i.e. there are no unknowns in the overlap. Then the cancellation still works, and AS is equivalent to the discretization of the parallel Schwarz method of Lions. Again the result of non-equivalence holds in general, see [34, Theorem 3.5]. This is a particularity of AS, it is not equivalent to the discretization of the parallel Schwarz method of Lions, a visual example is given in [34, Figure 3.2].

From our example, we see however also a very easy way to fix this problem: if we modify the restriction matrices used in the extension R_j^T to slightly modified $\tilde{R}_j^T$ matrices which correspond to a non-overlapping decomposition, for example for our simple model problem[5]

$$\tilde{R}_1:=\begin{bmatrix}1&&&&&\\&1&&&&\\&&1&&&\\&&&0&&\end{bmatrix},\quad \tilde{R}_2:=\begin{bmatrix}&&0&&&\\&&&1&&\\&&&&1&\\&&&&&1\end{bmatrix},$$

and then we use instead of (31) the iteration formula

$$\boldsymbol{u}^n=\boldsymbol{u}^{n-1}+(\tilde{R}_1^TA_1^{-1}R_1+\tilde{R}_2^TA_2^{-1}R_2)(\boldsymbol{f}-A\boldsymbol{u}^{n-1}), \tag{35}$$

[5] One could also use the original R_1 and put two zeros into R_2, or vice versa, or use any partition of unity such that $\tilde{R}_1^TR_1+\tilde{R}_2^TR_2=I$, the identity.

then the magic cancellation works again like in Multiplicative Schwarz (27), and the updating formula (34) becomes

$$\boldsymbol{u}^n = \begin{bmatrix} u^n_{1,1} \\ u^n_{1,2} \\ u^n_{1,3} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ u^n_{2,4} \\ u^n_{2,5} \\ u^n_{2,6} \end{bmatrix}. \tag{36}$$

This is called the Restricted Additive Schwarz method (RAS), which was invented in [9] because the authors observed that the method converged better than AS when used as a preconditioner (for preconditioning, see Section 7). We see that RAS is equivalent to the discretization of the parallel Schwarz method of Lions, and it took almost a decade to understand that this result not only holds just for simple examples, but in full generality, see [34, Theorem 3.7][6].

Like for Multiplicative Schwarz however, the RAS implementation (35) is more expensive than implementing directly the discretized parallel Schwarz method of Lions (32), since again one has to compute the residual, in addition to the subdomain solves. Also in a parallel computing environment, one has to communicate in RAS half of the overlap, which is better than in AS and Multiplicative Schwarz, but still much more costly than just the interface variables like in the discretized parallel Schwarz method of Lions. Nevertheless, RAS is still often implemented in practice, for convenience of coding, for example in PETSc, whose performance could thus be improved for production runs if this implementation was changed to an implementation of the discretized parallel Schwarz method of Lions. RAS is now the default preconditioner in PETSc, since it converges faster than AS also when used as a preconditioner. Its only drawback is that it is not symmetric, even when A is.

4 There are better transmission conditions than Dirichlet and Neumann

> *Our study here concerns a new variant of the Schwarz method, adapted to the situation of an arbitrary number of nonoverlapping subdomains.*
> Pierre Louis Lions [64], 1989.

We have seen in Section 2 that Schwarz methods use Dirichlet transmission conditions and overlap to converge, while all the other DD methods, the Dirichlet-Neumann, the Neumann-Neumann and the FETI method use both

[6] A further variant, Additive Schwarz with harmonic extension, where the $\tilde{R}_j$ are on the right instead of the left, can also be shown under certain conditions to be equivalent to the discretization of the parallel Schwarz method of Lions [61, 62]. Putting the $\tilde{R}_j$ operators both on the left and on the right is called RASH. RASH has no relation to discretized Schwarz methods, since no interface data is extracted and transmitted, and even as preconditioner has no good convergence properties.

Dirichlet and Neumann conditions, and converge without overlap. Naturally the question arises for Schwarz methods if one could not also use Neumann conditions, or even better combinations of Dirichlet and Neumann conditions. In order to get more insight, we consider again our simple 1D model Poisson problem (1), and an alternating Schwarz method that uses now instead of Dirichlet transmission conditions a combination of Dirichlet and Neumann,

$$\begin{array}{rlrl} \partial_{xx}u_1^n = f & \text{in } \Omega_1, & \partial_{xx}u_2^n = f & \text{in } \Omega_2,\\ u_1^n(0) = g_l, & & (-\partial_x+p_2)u_2^n(\alpha) = (-\partial_x+p_2)u_1^n(\alpha), & \\ (\partial_x+p_1)u_1^n(\beta) = (\partial_x+p_1)u_2^{n-1}(\beta), & & u_2^n(1) = g_r. & \end{array} \tag{37}$$

This new Schwarz methods thus uses so called Robin transmission conditions, and p_1 and p_2 are two parameters that can be chosen for fast convergence. Such Schwarz methods are called optimized Schwarz methods[7], because one determines the parameters by an optimization process. The error equations for the optimized alternating Schwarz method (37) are

$$\begin{array}{rlrl} \partial_{xx}e_1^n = 0 & \text{in } \Omega_1, & \partial_{xx}e_2^n = 0 & \text{in } \Omega_2,\\ e_1^n(0) = 0, & & (-\partial_x+p_2)e_2^n(\alpha) = (-\partial_x+p_2)e_1^n(\alpha), & \\ (\partial_x+p_1)e_1^n(\beta) = (\partial_x+p_1)e_2^{n-1}(\beta), & & e_2^n(1) = 0. & \end{array} \tag{38}$$

As for the classical Schwarz method, the solutions are simply linear functions,

$$e_1^n(x) = C_1^n x, \quad e_2^n(x) = C_2^n(1-x).$$

Inserting these functions into the transmission conditions in (38), we obtain

$$C_1^n(1+p_1\beta) = C_2^{n-1}(-1+p_1(1-\beta)), \quad C_2^n(1+p_2(1-\alpha)) = C_1^n(-1+p_2\alpha)).$$

Solving the first relation for C_1^n and inserting the result into the second relation yields

$$C_2^n = \frac{1-p_1(1-\beta)}{1+p_1\beta}\frac{1-p_2\alpha}{1+p_2(1-\alpha)}C_2^{n-1},$$

and therefore the convergence factor ρ of this optimized Schwarz method is

$$\rho = \frac{1-p_1(1-\beta)}{1+p_1\beta}\frac{1-p_2\alpha}{1+p_2(1-\alpha)}.$$

We see that for p_1 and p_2 large we recover the convergence factor of the classical alternating Schwarz method in (4), which is natural, since p_1 and p_2 large means using approximately Dirichlet transmission conditions. If we use $p_1 = p_2 = 0$, which corresponds to using Neumann transmission conditions, then the convergence factor is 1 and the method does not work, so using Neumann conditions in a Schwarz method is not a good idea. Using however a good combination of Neumann and Dirichlet is extremely powerful: if we choose $p_1 = p_1^* := \frac{1}{1-\beta}$

[7] A name introduced in [39], see also the review [33] for a more complete introduction.

and $p_2 = p_2^* = \frac{1}{\alpha}$, then the convergence factor becomes identically zero, and the optimized Schwarz method becomes a direct solver. This reminds us of the best choice of the relaxation parameter in Dirichlet-Neumann, Neumann-Neumann and FETI, but is possible here without relaxation. It turns out that this is a much more powerful approach than the relaxation parameter in the other methods, since optimized Schwarz methods can also become direct solvers for more than two subdomains using such transmission conditions, which is not possible for the other DD methods [13]. The optimal choice for the parameters is in general a Dirichlet to Neumann operator, see [33] for a simple introduction, and this led to the Schwarz methods by domain truncation [56]. Approximating such Dirichlet to Neumann operators for best convergence is a very active field of research and leads to Schwarz methods that are tuned to perform best for the given physical problem and PDE they are solving. The resulting optimized Schwarz methods are the best current solvers for the hard class of time harmonic wave propagation problems, like Helmholtz (see the seminal work by Després [24]), time harmonic Maxwell and elasticity, see [55] for more complete references. The concept of using Dirichlet to Neumann operators in transmission conditions is so fundamental that it has been independently reinvented by several research groups around the world, under different names, like the Sweeping Preconditioner [29, 30], the source transfer domain decomposition method [17, 18], the single layer potential method [73], and the method of polarized traces [77], and all these methods are mathematically optimized Schwarz methods, see [55]. Also the early approximate LU factorization methods AILU [47, 1, 48] are in this class. Historically, the idea of such transmission conditions goes back to [66] and even Lions [63] whose goal was to obtain a convergent non-overlapping Schwarz method, see the quote at the beginning of this section and [34] for a more comprehensive historical review.

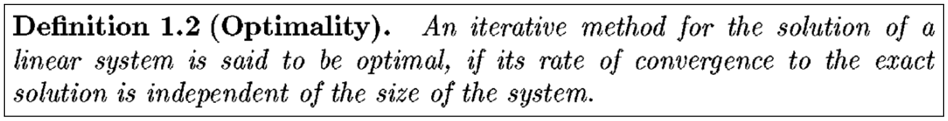
Using such transmission conditions which take the physical properties into account is also extremely important for heterogeneous domain decomposition problems, where the model in different subdomains is different. It is only with such transmission conditions that one can obtain the most advantageous convergence for a given problem, see e.g. [51], and convergence can even benefit form the fact that the physics are different in different subdomains, see e.g. [35].

5 On the meaning of 'optimal' in DD

> *"We establish the existence of interface conditions which are optimal in terms of iteration counts".*
>
> Dolean, Jolivet, Nataf [27], 2015.

We use the word 'optimal' quite naturally in common daily conversations across many languages. Its meaning according to standard dictionaries is

best or most favorable (Oxford Dictionaries),
best or most effective in a particular situation (Cambridge Dictionary),
most desirable or satisfactory (Merriam-Webster) .



Definition 1.2 (Optimality). *An iterative method for the solution of a linear system is said to be optimal, if its rate of convergence to the exact solution is independent of the size of the system.*

Fig. 16. Mathematical definition of 'optimal' in Domain Decomposition, taken from [74].

In the DD literature, with some important exceptions, see e.g. the quote above referring to what we have seen in Section 4 just before, there is however a different, precise mathematical definition of 'optimal', as shown in Figure 16. The word 'optimal' in the DD literature does therefore not mean 'best or most favorable' like in our common use of language. This led to many quite funny discussions among researchers in the more general field of iterative methods. For example a researcher in DD would say 'I have an optimal method', while another researcher from the multigrid community would say 'I have a faster method', to which the first would reply 'but mine is optimal' and the second would say 'mine is the fastest'. When I was a young professor at McGill University in Montreal, a German scientist once asked me with all the expertise in DD that I was supposed to have gained in the meantime, which DD methods were better, the ones with overlap or the ones without, and which one I would recommend for him to use for his application. And I was not able to answer: if we look at the seminal and fundamental condition number estimates in Section 1, they are all 'optimal' and one can not say which method is better really.

From the mathematical definition in Fig. 16, we can see that 'optimal' in reality means 'scalable', which has a clear meaning as a word and is a very important concept for iterative methods for solving partial differential equations. I would therefore like to propose to change this definition and replace 'optimal' by 'scalable'. Then we can have many scalable methods, some are faster and some are slower, and we can still try to search for the fastest scalable method, which I would then call 'optimal', since then the word would be used in the original sense 'best or most favorable' which we are used to.

Now scalability is a very important concept, and the size of the system that appears in the definition in Fig. 16 can be increased in two natural ways:

1. One keeps the domain fixed and refines the mesh; this typically happens if one wants to get more accuracy for a given simulation and is willing to pay for a larger computer.
2. One increases the domain by adding subdomains. This typically happens in applications where the domain is not fixed, for example molecular simulations where one wants to simulate larger and larger molecules.

In scalability one also distinguishes strong scalability, in which one keeps the problem (matrix) size fixed and wants to solve the problem faster and faster using more and more processors, and weak scalability, in which one wants to solve a larger and larger problem in the same time using more and more processors.

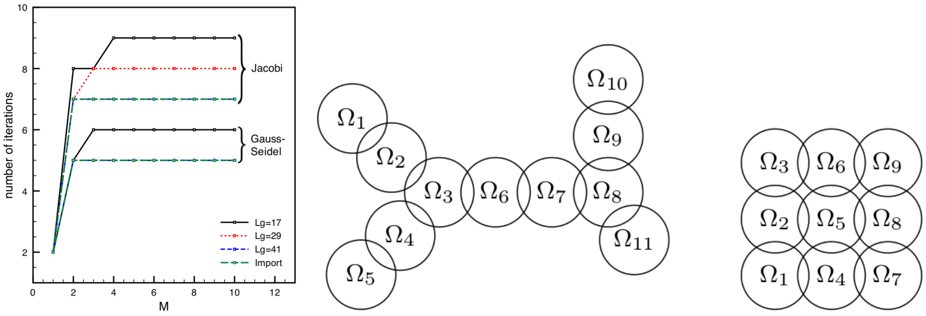


Fig. 17. Left: Scalability of the classical (one level) Schwarz method for a molecular simulation. Middle and right: configurations for which such scalability holds.

The desire for scalability in DD is thus the desire to have methods whose convergence rate does not deteriorate when one uses more and more processors, i.e. more and more subdomains, as in the definition in Fig. 16. In general, all domain decomposition methods we have seen in Section 2 are not scalable, as one can find in the DD literature, e.g.:

> "... we have shown that if we only have next neighbors communication, the minimum number of iterations required grows at least as fast as $N^{\frac{1}{2}}$, where N is the number of substructures." *Dryja and Widlund (1987)*
>
> "It is well known that the absence of a coarse problem results in deterioration of convergence of the iteration with increasing number of subdomains." *Mandel and Brezina (1993)*

There are however important exceptions, and I would like to mention three of them. The first one was discovered in [10] in a specific molecular dynamics simulation for a solvation model. If one added more and more atoms to a molecule, and each atom represented a subdomain, the classical one level Schwarz method converged with a number of iterations that did not increase, see Fig. 17 on the left. Jacobi in the figure refers here to the parallel Schwarz method of Lions, and Gauss-Seidel to the alternating Schwarz method, and M to the number of atoms (subdomains) in the molecule[8]. This surprising scalability of the one level Schwarz method was observed when solving a Poisson problem on domains composed of discs (or spheres) like the ones shown in the middle and on the right in Fig. 17 with Dirichlet boundary conditions at the outer boundary. The scalability is due to the Dirichlet boundary condition, with Neumann boundary

[8] In the literature one sometimes finds the name 'Jacobi Schwarz' method or 'Gauss-Seidel' Schwarz method, even I used this terminology early on, because at the discrete level we have seen the relation with block Jacobi and block Gauss-Seidel. But it is not appropriate, since neither Jacobi nor Gauss nor Seidel studied Schwarz methods or contributed to them. It is more appropriate just to talk about the parallel and alternating Schwarz method.

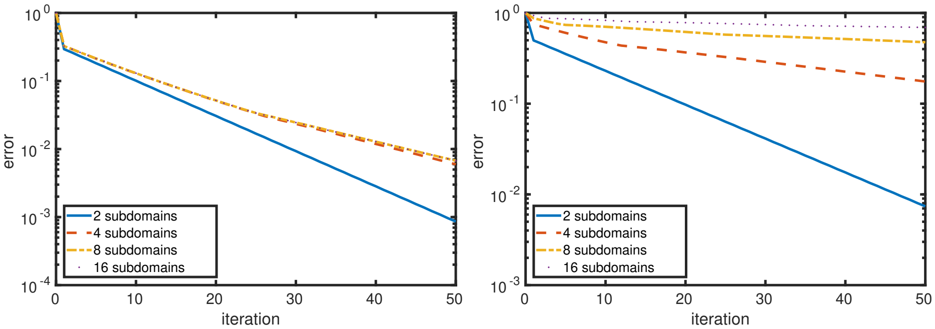


Fig. 18. One level Schwarz method for the $\eta - \Delta$ problem in 1D with increasing number of subdomains by adding subdomains of fixed size. Left: $\eta = 5$. Right: $\eta = 0$, the Laplace problem.

conditions it would not hold, and this was later proved using the three classical convergence analysis techniques for Schwarz methods: Fourier analysis [19] for convergence in L^2, the maximum principle [20] for convergence in L^∞, and the projection technique of Lions [21] for convergence in H^1. Note that on the right in Figure 17, the key ingredient are the holes with Dirichlet conditions in the domain for scalability. It was later also shown that all other one level DD methods we have seen in Section 2 are scalable in this situation [12].

Next, there are also one level DD methods which are scalable because of the method, a very interesting example is the method presented in [73], where the author says:

> "Generally, an increase in the number of subdomains leads to an increase in the number of iterations required for convergence. Here we propose and study a method where the number of iterations is essentially independent of the number of subdomains."

Here, the method is an optimal[9] alternating Schwarz method, equivalent to a block LU decomposition, and thus a direct solver, using Dirichlet to Neumann operators in the transmission conditions for a strip decomposition. It executes one forward sweep and one backward sweep, corresponding to the forward and backward solve in the block LU decomposition, and thus does not depend on the number of blocks, see [55] for more information.

Scalability can also be achieved because of the partial differential equation which is solved. For example problems of the form $(\eta - \Delta)u = f$ with $\eta > 0$ can be solved in a scalable fashion with one level DD methods when the domain grows with the number of subdomains, as is illustrated for a parallel Schwarz method in Fig. 18 on the left in 1D. We used here a decomposition of the domain $\Omega := (0, 1)$ decomposed into J overlapping subdomains $\Omega_j := (\alpha_j, \beta_j)$, $j = 1, 2, \ldots, J$, with

[9] Not in the sense of scalability, but really better is not possible, like in the quote at the beginning of this Section!

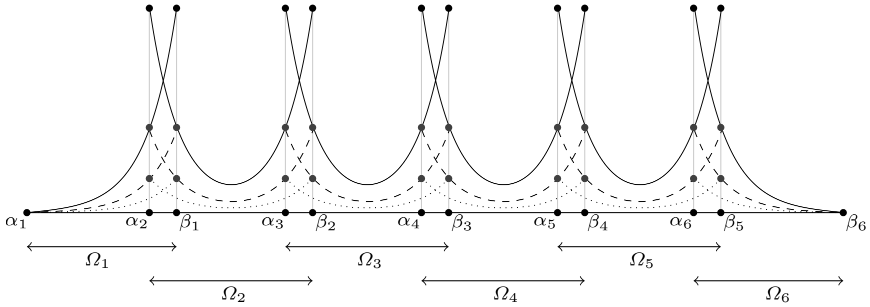


Fig. 19. Error of the parallel Schwarz method with $J = 6$ subdomains for the $\eta - \Delta$ problem with $\eta = 5$. The solid lines are the errors at the first iteration e_j^1, the dashed lines e_j^2, and the dotted lines are e_j^3.

$0 =: \alpha_1 < \alpha_2 < \beta_1 < \alpha_3 < \beta_2 < \ldots < \alpha_J < \beta_{J-1} < \beta_J := 1$. We see that when adding subdomains of the same size, i.e. increasing J but keeping $\beta_j - \alpha_j$ constant, convergence does not deteriorate when going from $J = 4$ to $J = 8$, and then to $J = 16$ subdomains. The reason for this can be easily understood by looking at the drawing in Fig. 19. We see that with an initial error equal to 1 (for other errors the situation would be similar), on each subdomain Ω_j, no matter if it is close or far away from the external boundaries of the overall domain $\Omega := (\alpha_1 = 0, \beta_6 = 1)$ the solution of the error equations in the interior of each subdomain is decaying as we move away from the interfaces at α_j and β_j, and thus the error is decaying uniformly in the classical one level parallel Schwarz method as the iteration progresses, independently of how many subdomains one has, provided the subdomains remain of the same size, like in the molecular simulation above. Hence the operator can provide the same decay as the Dirichlet boundary conditions in the molecular simulation and provide scalability, and this in any spatial dimension. Another example where this happens is when Schwarz waveform relaxation is applied to the second order time dependent wave equation on bounded time intervals $t \in (0, T)$, see for example [38, Theorem 3.5].

This is however very different if we consider the Laplace problem, i.e. the case with $\eta = 0$. Then, when one uses more and more subdomains in this example, DD methods converge more and more slowly, as shown in Fig. 18 on the right. This can again best be understood looking at how the error now decays in this case as the iterations progress. Assuming that we start again with an initial error equal to one on each interface, we obtain the sequence of error iterates shown in Fig. 20. We see that at the first iteration, only the errors on the subdomains that touch the outer boundary diminish. In the next iteration, the errors in the subdomains which touch a boundary domain diminish, and in the third iteration finally the subdomains in the middle also start to have smaller error. This indicates that if one has for example $J = 1000$ subdomains, at least $J/2 = 500$ iterations are needed before the subdomains in the middle are experiencing decay in the

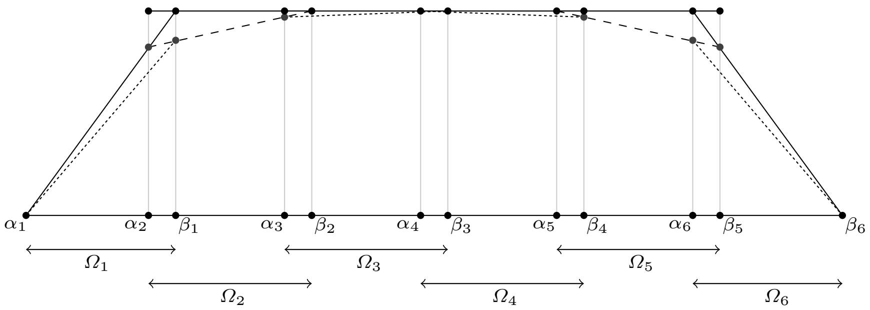


Fig. 20. Error of the parallel Schwarz method with $J = 6$ subdomains for the Laplace problem. The solid lines are the errors at the first iteration e_j^1, the dashed lines e_j^2 (only visible for $j = 2, 5$, the others are on top of the black lines), and the dotted lines are e_3^3 and e_4^3.

error, the method needs a number of iterations proportional to the number of subdomains.

What can one do in this case? To show this concretely, consider the discretized parallel Schwarz method with J equal subdomains for the 1D Poisson equation using centered finite differences, which gives a linear system of the form (26) but with more unknowns. The discretized parallel Schwarz method becomes a block Jacobi method, as we have seen in (23) for the special case of two subdomains, and if we use minimal overlap, such that the augmented system and the original system coincide, see Remark 1, the splitting $A = M - N$ in the iteration (21) would for example with three unknowns per subdomain be of the form

$$M := -\frac{1}{h^2}\left[\begin{array}{ccc|ccc|c} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & & & & \\ \hline & & & -2 & 1 & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 & \\ \hline & & & & & & \ddots \\ & & & & & & \quad\ddots \end{array}\right], \quad N = A - M.$$

We show in Fig. 21 first again how such a Schwarz method takes more and more iterations if one uses more and more subdomains, for the case of $J = 4$, $J = 8$ and $J = 16$ subdomains. This is a very undesirable property that all domain decomposition methods have in such cases, because they only communicate information between neighboring subdomains, and not globally, and the error within the subdomains does not have any natural decay coming either from the operator or the boundary conditions. As a remedy, one can add a coarse propagation mechanism, which can be done as follows: one adds to the basic matrix iteration

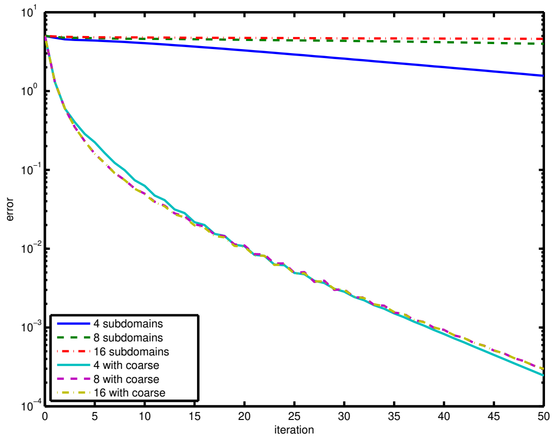


Fig. 21. Discretized parallel Schwarz with more and more subdomains, without and with coarse space correction.

(21) a second step to obtain a two step process,

$$\begin{aligned} \boldsymbol{u}^{n+\frac{1}{2}} &= \boldsymbol{u}^n + M^{-1}(\boldsymbol{f} - A\boldsymbol{u}^n), \\ \boldsymbol{u}^{n+1} &= \boldsymbol{u}^{n+\frac{1}{2}} + P(RAP)^{-1}R(\boldsymbol{f} - A\boldsymbol{u}^{n+\frac{1}{2}}), \end{aligned} \tag{39}$$

where the restriction matrix R and the prolongation matrix P for our example with three unknowns could be chosen for example as it was proposed by Nicolaides [68],

$$R := \frac{1}{3}\left[\begin{array}{c|c|c} 1\ 1\ 1 & & \\ \hline & 1\ 1\ 1 & \\ \hline & & \ddots \end{array}\right], \quad P := R^T. \tag{40}$$

We see that in the second step in (39), in the coarse correction, the restriction matrix R computes an average residual per subdomain, which is then corrected by a coarse matrix solve with the smaller matrix RAP which corrects the residual average simultaneously over all subdomains, and this correction is prolongated by P to the fine degrees of freedom and added to the current iterate $\boldsymbol{u}^{n+\frac{1}{2}}$ from the parallel Schwarz step. Using such a coarse space correction tremendously improves the convergence of the parallel Schwarz method, as one can see in Fig. 21, and makes it scalable, i.e. now the number of iterations does not depend on the number of subdomains any more.

All DD methods we have seen, Schwarz methods, Dirichlet-Neumann methods, Neumann-Neumann methods, and FETI methods become scalable when using such a coarse correction, i.e. the number of iterations does not grow any more when one increases the number of subdomains and thus the number of processors one can use. In the classical terminology of DD, one would thus call all these methods 'optimal', even though one can not say if one is better than the other, and naturally the question arises if one can not make these methods

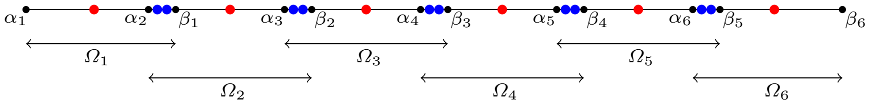


Fig. 22. Classical (red) and new location (blue) of the coarse space nodes within the overlap using piecewise linear coarse functions of the parallel Schwarz method.

even faster, i.e. are there no better coarse spaces than just using the constant average per subdomain? Often one uses piecewise linear coarse spaces which already work much better, but much more is possible, as we will see next.

6 Better than 'optimal' in the classical DD sense

"We approximate the optimal operators".
Dolean, Jolivet, Nataf [27], 2015.

To see how the coarse correction acts concretely in such a parallel Schwarz method, it is best to look at a simulation. We use again the Laplace problem in 1D and $J = 4$ subdomains. We show in Fig. 23 how the iterations of the two level Schwarz method converge when using a piecewise linear coarse space with one degree of freedom in the center of each subdomain, as indicated in red in Fig. 22. We see that the piecewise coarse correction is very effective in the first iteration to transport the solution instantly from the outer boundary to subdomains that do not touch the boundary. We see however also that with the coarse nodes in the middle of the subdomains, the coarse correction destroys the good property that the subdomain approximations were solutions of the subdomain problems, a fact that was first pointed out in [40]. This leads in the second parallel subdomain solve to important corrections that the subdomains must bring to the iterate, which are then again spread by the coarse solve which breaks again the subdomain solution property and so on. The method is 'optimal' in the sense of scalable, but there is clear potential to do a better coarse correction.

A first simple idea is to use a coarse space which is not destroying the subdomain solution property, i.e a coarse space which only uses functions that are themselves solutions of the homogeneous problem in the subdomain, in our example piecewise linear throughout the subdomains, i.e. harmonic, solutions of the homogeneous problem to be solved. This implies that the coarse mesh points should lie within the overlap only, as indicated in Fig. 22 with the blue coarse mesh points. If we use precisely the same code as before, but just the new placement of the coarse nodes, then we obtain the parallel Schwarz iterates shown in Fig. 24. We see that this coarse space is as effective to transport information from the outer boundaries to the interior subdomains as the one using the red points in the center of the subdomains, but even more, it does not destroy the solution properties of the subdomain solutions, i.e. after the first coarse correction the solution is obtained in all subdomains, only a small error remains in the overlap, which is easily corrected with one more subdomain solve, and

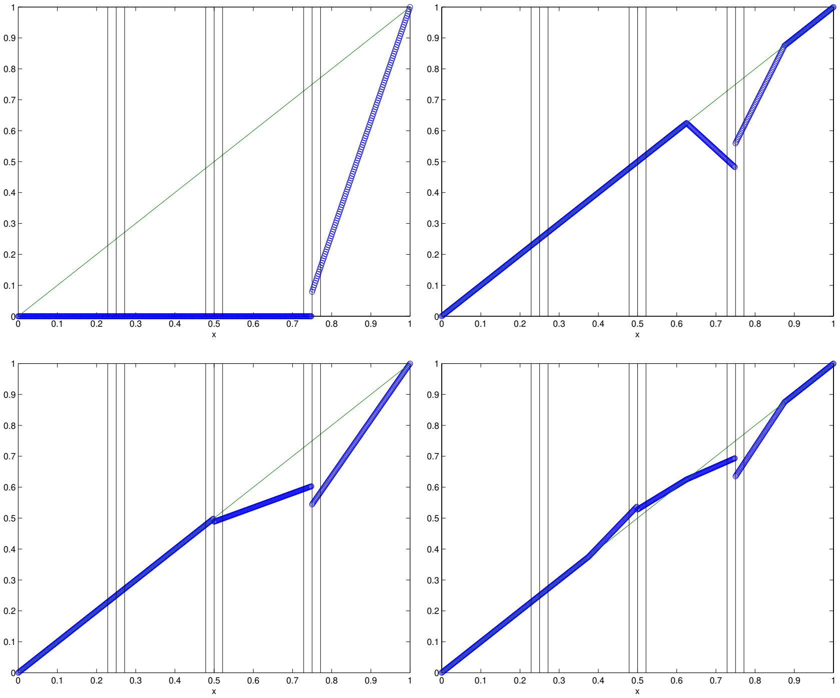


Fig. 23. Two level parallel Schwarz method iterates with piecewise linear coarse correction and a coarse degree of freedom in the center of each subdomain. Top: after the first parallel subdomain solve and after the first course correction. Bottom: after the second parallel subdomain solve and after the second course correction.

the following second coarse correction is identically zero. This two level Schwarz method is thus becoming a direct solver, like if one had factored the problem in a direct solver.

In higher spatial dimensions, such a coarse space which makes the parallel Schwarz method into a direct solver would need for the Poisson problem to contain all harmonic functions in each subdomain, and would thus be infinite dimensional, see [44], and one thus has to use approximations, and decide which harmonic functions to use for the coarse space. To do so, with the computing power we have currently available, one can just look at which functions would be important, i.e. which functions remain in the error when running the parallel Schwarz method, or in other words the eigenfunctions of the Schwarz iteration operator with largest eigenvalue in modulus. We show these eigenfunctions in Figure 25, for a decomposition of the unit square domain into 8×8 equal square subdomains. We see that the eigenfunctions of the Schwarz operator appear in pairs, always a continuous and a discontinuous one, and they look like the eigenfunctions of the Laplace operator, except that they are piecewise harmonic within subdomains. This shows that also in 2D the coarse space should con-

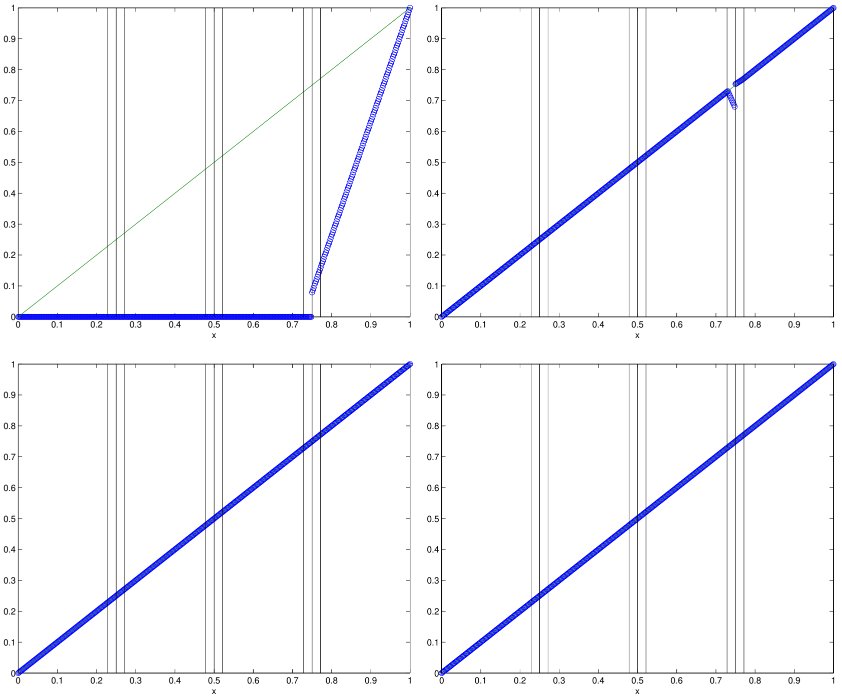


Fig. 24. Two level parallel Schwarz method iterates with piecewise linear coarse correction and the coarse degrees of freedom in the overlap. Top: after the first parallel subdomain solve and after the first course correction. Bottom: after the second parallel subdomain solve.

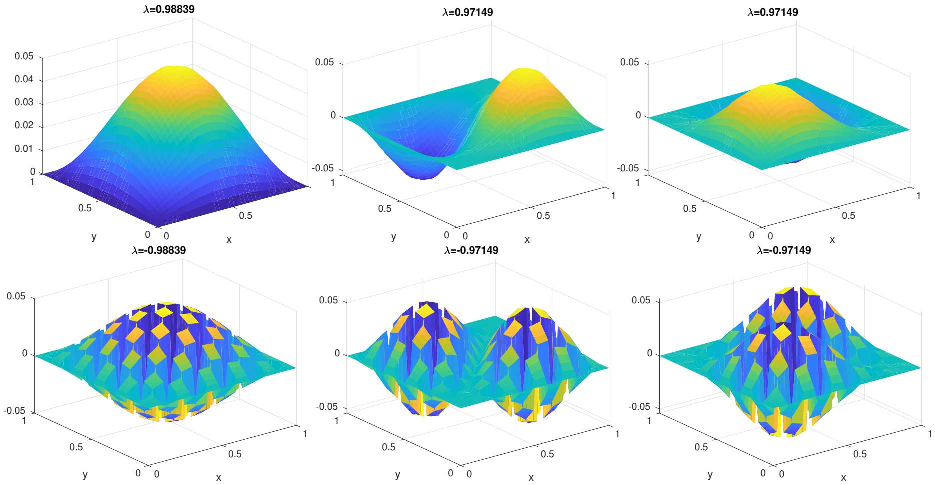


Fig. 25. Eigenfunctions of the parallel Schwarz method with 8 × 8 subdomains on the unit square.

tain harmonic functions within subdomains, in this example $Q1$ finite element functions aligned with the square subdomain decomposition would be ideal to span these functions, and if one wants to remove both the continuous and discontinuous coarse functions the coarse space would need to be discontinuous. Looking however also at the corresponding eigenvalues indicated above each eigenfunction, we see that the discontinuous Schwarz eigenfunctions always have the eigenvalue with a negative sign corresponding to the continuous one with the positive sign. This is precisely what happens in the point Jacobi smoother when using multigrid [58]: there is always a smooth and a corresponding highly oscillatory eigenfunction with the same eigenvalue in magnitude, but different sign. One therefore should use, like in multigrid, the Schwarz iteration with a damping parameter $\frac{1}{2}$, or even better $\frac{2}{3}$, see for example [22, Section 4.10], and then the oscillatory eigenfunctions are damped very quickly, and the coarse correction only needs to correct the smooth ones. All these properties are not visible when one uses the Schwarz method directly as a preconditioner with conjugate gradients (CG), and they do also not appear in condition number estimates, even though the important link with multigrid is well known in the DD literature, see e.g. [75].

Using such a coarse space and also optimized transmission conditions we have seen in Section 4, Schwarz methods just used as iterative solvers, and not as preconditioners for CG, can become competitive with multigrid solvers for Poisson problems, which are known to be the best iterative solvers for such problems not needing any CG or Krylov acceleration, see Figure 26, taken from [50]. Such Schwarz methods are now much closer to ‘optimal’ in the sense we use the word optimal in common language use, as explained in Section 5. But can one make them even faster? The question is if one could put more coarse

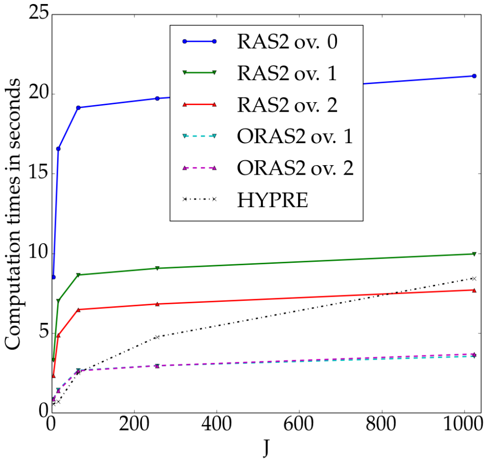


Fig. 26. Comparison in PETSc of the parallel Schwarz method with $Q1$ coarse correction and classical Dirichlet transmission conditions (RAS) and optimized transmission conditions (ORAS) and different (algebraic) overlap with HYPER for a growing number of subdomains J in 2D taken from [50].

functions in addition to the $Q1$ functions, especially if the coarse problem is still much much smaller than the subdomain problems, which can be arbitrarily expensive when the mesh is refined for more accuracy.

The answer is yes, it suffices to look at the eigenfunctions of the Schwarz operator that come after all the $Q1$ functions per subdomain that look like the eigenfunctions of the Laplacian. To do so, we show in Fig. 27 for a simple 2×2 decomposition the corresponding eigenfunctions that also come later. Due to the symmetry of the decomposition, we only show the eigenfunctions that concentrate on the interface $(0,1) \times \{\frac{1}{2}\}$, the eigenfunctions on the other interface $\{\frac{1}{2}\} \times (0,1)$ are the same, just rotated by 90 degrees. We also show only the continuous eigenfunctions, since the discontinuous ones are just obtained by flipping the sign, for example in two diagonal subdomains. We see that after the $Q1$ hat function in the top left of Figure 27, the eigenfunctions concentrate on the interface and become more and more oscillatory as their corresponding eigenvalue is decreasing. We also see that there are two groups. The first one is in plots 2, 4, 6 and 8 from top left to bottom right. These are simply the sine functions along the interface $(0,1) \times \frac{1}{2}$, extended harmonically to the subdomains, as used in the SHEM coarse space [44, 43, 2], and which are very much related to the Dirichlet to Neumann map coarse spaces [67] and the GenEO coarse space [72], see [36]. The second group is in the plots 1 (yes already!), 3, 5, 7 and 9, and has at the cross point the characteristic peak already visible in the first $Q1$ coarse function at the top left. This class of coarse functions has so far been missing in all modern spectral coarse spaces, and was only discovered very recently, see [23], with

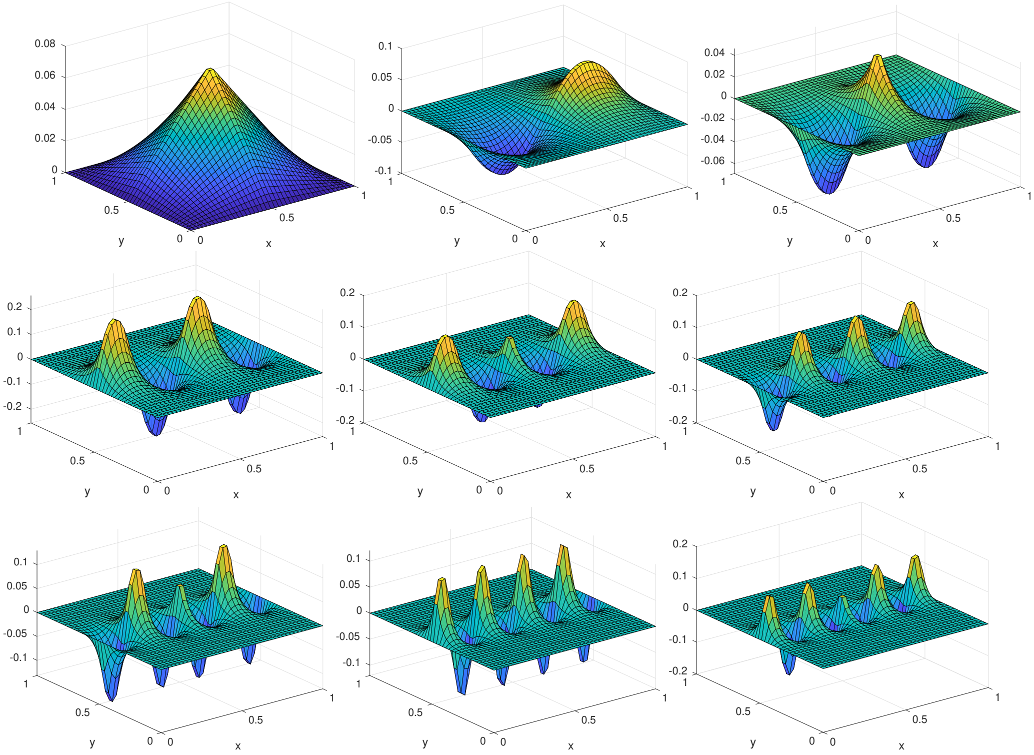


Fig. 27. Continuous eigenfunctions of the Schwarz operator for the Laplacian for a 2×2 decomposition.

closed form expressions for all these coarse functions at the continuous level. A systematic construction for assembling coarse spaces based on these coarse functions that works extremely well, also for general decompositions using Metis for example, and also with optimized transmission conditions, can also be found in [23]. These coarse space functions, which are naturally indicating that the cross points in DD methods are special, can neither be detected nor computed using other spectral coarse space techniques, and are also very interesting for non-overlapping DD methods like the Neumann-Neumann method [14], which is not well posed at the continuous level in the presence of cross-points, an issue that has only been understood very recently [15], and the same holds for the Dirichlet-Neumann method [16]. Cross-points need in general special treatment if one wants to obtain best performance, not just scalability, both in theoretical convergence proofs [42] as well as when discretizing Domain Decomposition methods [49, 37] and optimizing transmission conditions [41].

Note that modern spectral coarse spaces like GenEO [67], SHEM [44], the ACMS coarse space [60], and also new versions of the GDSW coarse space [59] were not developed in order to obtain a better DD method for solving Laplace problems, but to handle the difficult case of high contrast problems. As we have seen here however, such coarse spaces would also be obtained applying the presented spectral computation for the high contrast problem, and we would then not just thrive for scalability, but truly best performance, or 'optimality' in the sense the word is used in common language.

7 Preconditioning

> *"In the past, Krylov space solvers were referred to also by other names such as semi-iterative methods and polynomial acceleration methods".*
> Gutknecht [57], 2007.

We have so far considered all DD methods like classical stationary iterative solvers, which solve problems only on subdomains, and then exchange information with their neighbors, or further via a coarse space, and converge to the solution of the underlying problem, except in Section 1, where we saw that the most sophisticated historical DD Solvers were all preconditioners, not iterative solvers. It is now time to explain why this is important in practice when using DD methods, while for their understanding it is easier to first studying them in their stationary iterative form, as we have seen in Sections 2 to 6.

As we have seen in Section 3, continuous DD methods when discretized can all be written as stationary iterations of the form (21) with $A = M - N$. Such stationary iterations can be rewritten in residual correction form,

$$\boldsymbol{u}^n = M^{-1}(N\boldsymbol{u}^{n-1} + \boldsymbol{f}) = M^{-1}((M-A)\boldsymbol{u}^{n-1} + \boldsymbol{f}) = \boldsymbol{u}^{n-1} + M^{-1}(\boldsymbol{f} - A\boldsymbol{u}^{n-1}), \tag{41}$$

where now the preconditioned residual $\boldsymbol{r}^n := M^{-1}(\boldsymbol{f} - A\boldsymbol{u}^n)$ appears. At convergence, when $\boldsymbol{u}^n$ converges to $\boldsymbol{u}$, we get from (41)

$$\boldsymbol{u} = \boldsymbol{u} + M^{-1}(\boldsymbol{f} - A\boldsymbol{u}) \quad \Longleftrightarrow \quad M^{-1}A\boldsymbol{u} = M^{-1}\boldsymbol{f},$$

the so called preconditioned system, where M is the preconditioner. Such systems one often solves with a Krylov method, instead of using the stationary iteration (41), because convergence is much faster. For example GMRES (Generalized Minimum RESidual) would find at iteration n an approximate solution $\boldsymbol{u}^n_{\text{kry}} \in \boldsymbol{u}_0 + \text{span}\{\boldsymbol{r}_0, M^{-1}A\boldsymbol{r}_0, (M^{-1}A)^2\boldsymbol{r}_0, \dots, (M^{-1}A)^{n-1}\boldsymbol{r}_0\}$, the so called affine Krylov space, such that $||r^n_{\text{kry}}||_2 := ||M^{-1}\boldsymbol{f} - M^{-1}A\boldsymbol{u}^n_{\text{kry}}||_2$ is minimized. As the following theorem shows, using GMRES is an accelerator for such stationary iterations.

Theorem 1. *Consider a linear system $A\boldsymbol{u} = \boldsymbol{f}$ and the matrix splitting $A = M - N$ with M invertible and the corresponding approximate solutions $\boldsymbol{u}^n$ of the stationary method* (21) *and the approximate solutions $\boldsymbol{u}^n_{kry}$ obtained by GMRES and the corresponding two preconditioned residuals*

$$\boldsymbol{r}^n := M^{-1}(\boldsymbol{f} - A\boldsymbol{u}^n), \quad \boldsymbol{r}^n_{kry} := M^{-1}(\boldsymbol{f} - A\boldsymbol{u}^n_{kry}).$$

Then $||\boldsymbol{r}^n_{kry}||_2 \leq ||\boldsymbol{r}^n||_2$ for $n = 1, 2, \dots$.

Proof. First, notice that $\boldsymbol{r}^0 = \boldsymbol{r}^0_{\text{kry}}$ since we use the same initial guess $\boldsymbol{u}^0$. Then, the stationary iterative method based on the splitting $A = M - N$ computes at the iteration n, as we have seen in (41), the approximation $\boldsymbol{u}^n = \boldsymbol{u}^{n-1} + \boldsymbol{r}^{n-1}$, where the *preconditioned residual* $\boldsymbol{r}^n$ satisfies

$$\begin{aligned}\boldsymbol{r}^n &= M^{-1}\boldsymbol{f} - M^{-1}A\boldsymbol{u}^n\\ &= M^{-1}\boldsymbol{f} - M^{-1}A(\boldsymbol{u}^{n-1} + \boldsymbol{r}^{n-1})\\ &= \boldsymbol{r}^{n-1} - M^{-1}A\boldsymbol{r}^{n-1} = (I - M^{-1}A)\boldsymbol{r}^{n-1}.\end{aligned}$$

By induction, we thus get

$$\boldsymbol{r}^n = (I - M^{-1}A)^n\boldsymbol{r}^0.$$

Therefore, we have with the residual polynomial $p^n(x) := (1-x)^n$, $p^n(0) = 1$ that

$$\boldsymbol{r}^n = p^n(M^{-1}A)\boldsymbol{r}^0.$$

GMRES on the other hand finds at iteration n an approximation $\boldsymbol{u}^n_{\text{kry}}$ in the affine Krylov space

$$\boldsymbol{u}^0 + \operatorname{span}\{\boldsymbol{r}^0, M^{-1}A\boldsymbol{r}^0, \ldots, (M^{-1}A)^{n-1}\boldsymbol{r}^0\},$$

which means

$$\boldsymbol{u}^n_{\text{kry}} = \boldsymbol{u}_0 + \sum_{j=1}^{n}\gamma_j(M^{-1}A)^{j-1}\boldsymbol{r}^0,$$

with coefficients γ_j such that $||\boldsymbol{r}^n_{\text{kry}}||_2$ is minimized. The *preconditioned residual* $\boldsymbol{r}^n_{\text{kry}}$ of GMRES can be written as

$$\boldsymbol{r}^n_{\text{kry}} = M^{-1}\boldsymbol{f} - M^{-1}A\boldsymbol{u}^n_{\text{kry}} = M^{-1}\boldsymbol{f} - M^{-1}A\boldsymbol{u}_0 - M^{-1}A\sum_{j=1}^{n}\gamma_j(M^{-1}A)^{j-1}\boldsymbol{r}^0,$$

and hence

$$\boldsymbol{r}^n_{\text{kry}} = p^n_{\text{kry}}(M^{-1}A)\boldsymbol{r}^0, \text{ with } p^n_{\text{kry}}(0) = 1,$$

where p^n_{kry} is also a residual polynomial of degree lower than or equal to n. Now GMRES minimizing the residual will find the polynomial p^n_{kry} such that $\|\boldsymbol{r}^n_{\text{kry}}\|_2$ is as small as possible, and thus at least as small as $\|\boldsymbol{r}^n\|_2$ of the stationary iterative method which used the simple polynomial $p^n(x) = (1-x)^n$.

In general, GMRES gives $\|\boldsymbol{r}^n_{\text{kry}}\|_2 \ll \|\boldsymbol{r}^n\|_2$, so in practice one should always use Krylov acceleration for stationary methods of the form (21).

If the matrix A is symmetric and positive definite, and M as well, for example for Schwarz methods applied to the Poisson problem with minimal overlap[10], a

[10] With more than minimal overlap, the Schwarz method is not symmetric any more, even if the problem is symmetric. One can use Additive Schwarz instead then, which is symmetric, but commits an error in the overlap as we have seen in Section 3, and convergence is slower due to this [9].

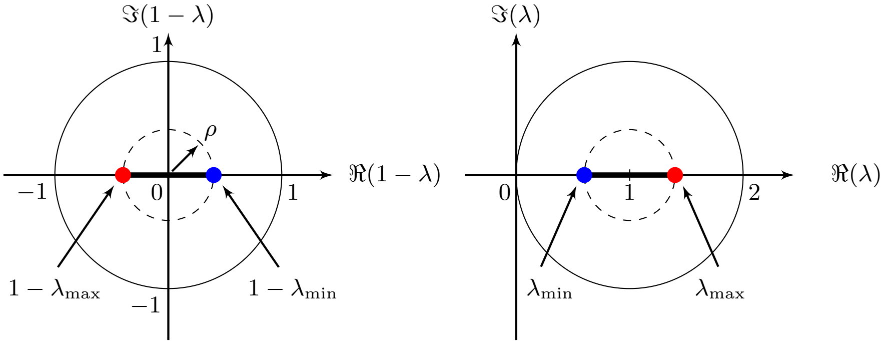


Fig. 28. Left: Preconditioner M in a stationary iteration, with spectral radius $\rho(I - M^{-1}A) = 0.4 < 1$. Right: corresponding use of the preconditioner M for a Krylov method with a real spectrum from $\lambda_{\min}(M^{-1}A)$ to $\lambda_{\max}(M^{-1}A)$.

similar optimality result like for GMRES above is also known for CG. For solving a linear system $A\boldsymbol{u} = \boldsymbol{f}$, CG finds at iteration n an approximate solution $\boldsymbol{u}_{\text{CG}}^n$ such that the error is minimized in the energy norm,

$$||\boldsymbol{e}_{\text{CG}}^n||_A \to \min,$$

and there is a convergence estimate of the form

$$||\boldsymbol{e}_{\text{CG}}^n||_A \leq 2\left(\frac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1}\right)^n ||\boldsymbol{e}_{\text{CG}}^0||_A.$$

Here $\kappa_2(A) := ||A||_2||A^{-1}||_2 = \lambda_{\max}(A)/\lambda_{\min}(A)$ denotes the two norm condition number of the matrix A, and $\lambda_{\max}(A)$ is the largest eigenvalue of A and $\lambda_{\min}(A)$ is the smallest eigenvalue of A. It is therefore interesting in this symmetric positive definite case to try to find preconditioners M such that the condition number of the preconditioned system $\kappa_2(M^{-1}A)$ is not too large[11].

This brings us to a very important point between solving a preconditioned system using a Krylov method and using the preconditioner in a stationary iteration. We have seen in (25) that for a stationary iteration to converge fast, the spectral radius of the iteration matrix must be small, $\rho(I - M^{-1}A) \ll 1$. When using CG, we want that the condition number of the preconditioned system $\kappa_2(M^{-1}A)$ is not too large. These two requirements are related, as is illustrated in Fig. 28. On the left we see that if the spectral radius $\rho(I - M^{-1}A)$ of the iteration matrix is smaller than one, all eigenvalues must lie in the small circle, and if they are real, they must lie between $1-\lambda_{\max}(M^{-1}A)$ and $1-\lambda_{\min}(M^{-1}A)$ on the real line. On the right, we see that when this same matrix M from the matrix splitting $A = M - N$ is used as a preconditioner, and the spectrum

[11] Note that the preconditioned system $M^{-1}A$ is not symmetric any more, but this can be handled elegantly in CG, see e.g. [22, Theorem 4.2].

is real, it must lie in the interval $(\lambda_{\min}(M^{-1}A), \lambda_{\max}(M^{-1}A))$. So if we made the spectral radius $\rho(I - M^{-1}A)$ on the left small to get fast convergence of the stationary iterative method, then the same preconditioner on the right is good for CG, since the condition number which is the ratio of the largest to the smallest eigenvalue, can not be large. More generally, if the spectral radius $\rho(I - M^{-1}A)$ on the left is small, all eigenvalues lie in the small circle, which means on the right all eigenvalues still lie in the small circle, close to one, and the Krylov method can easily find a residual polynomial $p^n(x)$ with $p^n(0) = 1$ at the origin, which is small on the small circle. So optimizing the method defined by M by making the spectral radius small leads in general also to M with good preconditioning properties for Krylov methods.

It is also important here to make a further more profound comment about Krylov methods, in context with preconditioning, and it is best to quote from the very recent reference [11]:

> “The nonlinear adaptivity of CG to the location of the individual eigenvalues indicates that a smaller condition number does not necessarily lead to faster convergence (contrary to widespread misinterpretations of the condition number bound in the literature). Therefore it is not recommended to use the minimization of the condition number as the only criterion for the choice of preconditioners. Alternatives to the condition number exist, but they require deep knowledge of CG and of the problem to be solved.”

While classical research in DD focused almost exclusively on condition number estimates, see Section 1, it could be very fruitful to shift the focus like we did in Section 6 to get a better understanding of the entire spectrum of the DD iteration operator, and possibly even more, in order to harvest all the knowledge in Krylov methods for obtaining truly best performing parallel methods based on DD, getting closer and closer to the word ’optimal’ in the common sense of spoken language.

Conclusions

Make things as simple as possible, but no simpler. Albert Einstein.

We gave a very personal view about DD, which is a very active field of research. Naturally, very important topics had to be left out, like for example heterogeneous DD methods (for a recent overview, see [46], and [51] for general elliptic problems), or DD methods for systems of partial differential equations like Maxwell’s equations [25, 5, 26] or elasticity [8], or time dependent problems (see the recent book [45] and review [54]). It is the hope of the author that his personal experience in this exciting and rapidly developing field of research will help young scientists to find their way more easily, and to be able to contribute rapidly with new and interesting results.

References

1. Y. Achdou and F. Nataf. Dimension-wise iterated frequency filtering decomposition. *Numer. Lin. Alg. Appl.*, 41:1643–1681, 2003.
2. P. E. Bjørstad, M. J. Gander, A. Loneland, and T. Rahman. Does SHEM for additive Schwarz work better than predicted by its condition number estimate? In *Domain Decomposition Methods in Science and Engineering XXIV 24*, pages 129–137. Springer, 2018.
3. P. E. Bjørstad and O. B. Widlund. Solving elliptic problems on regions partitioned into substructures. In *Elliptic Problem Solvers II*, pages 245–256. Academic Press, New York, 1984.
4. P. E. Bjørstad and O. B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, 23(6):1097–1120, 1986.
5. M. E. Bouajaji, V. Dolean, M. J. Gander, and S. Lanteri. Optimized Schwarz methods for the time-harmonic Maxwell equations with damping. *SIAM J. Sci. Comp.*, 34(4):A2048–A2071, 2012.
6. J.-F. Bourgat, R. Glowinski, P. Le Tallec, and M. Vidrascu. Variational formulation and algorithm for trace operator in domain decomposition calculations. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Domain Decomposition Methods. Second International Symposium on Domain Decomposition Methods*, pages 3–16, Philadelphia, PA, 1989. SIAM. Los Angeles, California, January 14–16, 1988.
7. J. H. Bramble, J. E. Pasciak, and A. H. Schatz. An iterative method for elliptic problems on regions partitioned into substructures. *Math. of Comp.*, 46(174):361–369, 1986.
8. R. Brunet, V. Dolean, and M. J. Gander. Natural domain decomposition algorithms for the solution of time-harmonic elastic waves. *SIAM J. Sci. Comp.*, 42(5):A3313–A3339, 2020.
9. X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comp.*, 21(2):792–797, 1999.
10. E. Cances, Y. Maday, and B. Stamm. Domain decomposition for implicit solvation models. *The Journal of chemical physics*, 139(5), 2013.
11. E. Carson, J. Liesen, and Z. Strakoš. Towards understanding CG and GMRES through examples. *Linear Algebra and its Applications*, 692:241–291, 2024.
12. F. Chaouqui, G. Ciaramella, M. J. Gander, and T. Vanzan. On the scalability of classical one-level domain-decomposition methods. *Vietnam Journal of Mathematics*, 46:1053–1088, 2018.
13. F. Chaouqui, M. J. Gander, and K. Santugini-Repiquet. On nilpotent subdomain iterations. In *Domain Decomposition Methods in Science and Engineering XXIII*, pages 125–133. Springer, 2017.
14. F. Chaouqui, M. J. Gander, and K. Santugini-Repiquet. A local coarse space correction leading to a well-posed continuous Neumann-Neumann method in the presence of cross points. In *Domain Decomposition Methods in Science and Engineering XXV 25*, pages 83–91. Springer, 2020.
15. B. Chaudet-Dumas and M. J. Gander. Cross-points in the Neumann-Neumann method. In *International Conference on Domain Decomposition Methods*, pages 117–124. Springer, 2022.
16. B. Chaudet-Dumas and M. J. Gander. Cross-points in the Dirichlet-Neumann method I: well-posedness and convergence issues. *Numer. Alg.*, 92(1):301–334, 2023.

17. Z. Chen and X. Xiang. A source transfer domain decomposition method for Helmholtz equations in unbounded domain. *SIAM J. Numer. Anal.*, 51:2331–2356, 2013.
18. Z. Chen and X. Xiang. A source transfer domain decomposition method for Helmholtz equations in unbounded domain Part II: Extensions. *Numer. Math. Theor. Meth. Appl.*, 6:538–555, 2013.
19. G. Ciaramella and M. J. Gander. Analysis of the parallel Schwarz method for growing chains of fixed-sized subdomains: Part I. *SIAM J. Numer. Anal.*, 55(3):1330–1356, 2017.
20. G. Ciaramella and M. J. Gander. Analysis of the parallel Schwarz method for growing chains of fixed-sized subdomains: Part II. *SIAM J. Numer. Anal.*, 56(3):1498–1524, 2018.
21. G. Ciaramella and M. J. Gander. Analysis of the parallel Schwarz method for growing chains of fixed-sized subdomains: Part III. *ETNA*, 49:201–243, 2018.
22. G. Ciaramella and M. J. Gander. *Iterative methods and preconditioners for systems of linear equations.* SIAM, 2022.
23. F. Cuvelier, M. J. Gander, and L. Halpern. Fundamental coarse space components for Schwarz methods with crosspoints. In *International Conference on Domain Decomposition Methods XXVI*, pages 39–50. Springer, 2023.
24. B. Després. *Méthodes de décomposition de domaine pour les problémes de propagation d'ondes en régime harmonique.* PhD thesis, Université Dauphine, Paris IX, 1991.
25. V. Dolean, M. J. Gander, and L. Gerardo-Giorda. Optimized Schwarz methods for Maxwell's equations. *SIAM J. Sci. Comp.*, 31(3):2193–2213, 2009.
26. V. Dolean, M. J. Gander, and E. Veneros. Asymptotic analysis of optimized Schwarz methods for Maxwell's equations with discontinuous coefficients. *ESAIM: Math. Mod. and Numer. Anal.*, 52(6):2457–2477, 2018.
27. V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods: algorithms, theory, and parallel implementation.* SIAM, 2015.
28. M. Dryja and O. B. Widlund. An additive variant of the Schwarz alternating method for the case of many subregions. Technical Report 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute, 1987.
29. B. Engquist and L. Ying. Sweeping preconditioner for the Helmholtz equation: Hierarchical matrix representation. *Comm. Pure Appl. Math.*, LXIV:0697–0735, 2011.
30. B. Engquist and L. Ying. Sweeping preconditioner for the Helmholtz equation: Moving perfectly matched layers. *Multiscale Model. Sim.*, 9:686–710, 2011.
31. C. Farhat and F.-X. Roux. A method of Finite Element Tearing and Interconnecting and its parallel solution algorithm. *Int. J. Numer. Meth. Engrg.*, 32:1205–1227, 1991.
32. D. Funaro, A. Quarteroni, and P. Zanolli. An iterative procedure with interface relaxation for domain decomposition methods. *SIAM J. on Num. Anal.*, 35(6):1213–1236, 1988.
33. M. J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731, 2006.
34. M. J. Gander. Schwarz methods over the course of time. *ETNA*, 31(5):228–255, 2008.
35. M. J. Gander and O. Dubois. Optimized Schwarz methods for a diffusion problem with discontinuous coefficient. *Numer. Alg.*, 69(1):109–144, 2015.

36. M. J. Gander and L. Halpern. Piece-wise constant, linear and oscillatory: a historical introduction to spectral coarse spaces with focus on Schwarz methods. In *Domain Decomposition Methods in Science and Engineering XXVII*, pages 189–198. Springer, 2022.
37. M. J. Gander and L. Halpern. A simple finite difference discretization for Ventcell transmission conditions at cross points. In *Domain Decomposition Methods in Science and Engineering XXVI*, pages 257–264. Springer, 2023.
38. M. J. Gander, L. Halpern, and F. Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. *SIAM J. Numer. Anal.*, 41(5):1643–1681, 2003.
39. M. J. Gander, L. Halpern, F. Nataf, et al. Optimized Schwarz methods. In *Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan*, pages 15–28, 2001.
40. M. J. Gander, L. Halpern, and K. S. Repiquet. A new coarse grid correction for RAS/AS. In *Domain Decomposition Methods in Science and Engineering XXI*, pages 275–283. Springer, 2014.
41. M. J. Gander and F. Kwok. Best Robin parameters for optimized Schwarz methods at cross points. *SIAM J. Sci. Comp.*, 34(4):A1849–A1879, 2012.
42. M. J. Gander and F. Kwok. On the applicability of Lions' energy estimates in the analysis of discrete optimized Schwarz methods with cross points. In *Domain decomposition methods in science and engineering XX*, pages 475–483. Springer, 2013.
43. M. J. Gander and A. Loneland. SHEM: An optimal coarse space for RAS and its multiscale approximation. In *Domain decomposition methods in science and engineering XXIII*, pages 313–321. Springer, 2017.
44. M. J. Gander, A. Loneland, and T. Rahman. Analysis of a new harmonically enriched multiscale coarse space for domain decomposition methods. *arXiv preprint arXiv:1512.05285*, 2015.
45. M. J. Gander and T. Lunet. *Time Parallel Time Integration*. SIAM, 2024.
46. M. J. Gander and V. Martin. An introduction to heterogeneous domain decomposition methods for multi-physics problems. In *Domain decomposition methods in science and engineering XXVII*, pages 215–222. Springer, 2023.
47. M. J. Gander and F. Nataf. AILU: a preconditioner based on the analytic factorization of the elliptic operator. *Numer. Lin. Alg. Appl.*, 7:505–526, 2000.
48. M. J. Gander and F. Nataf. An incomplete LU preconditioner for problems in acoustics. *J. Comp. Acoust.*, 13:455–476, 2005.
49. M. J. Gander and K. Santugini-Repiquet. Cross-points in domain decomposition methods with a finite element discretization. *ETNA*, 45:219–240, 2016.
50. M. J. Gander and S. Van Criekingen. New coarse corrections for optimized restricted additive Schwarz using PETSc. In *Domain Decomposition Methods in Science and Engineering XXV 25*, pages 483–490. Springer, 2020.
51. M. J. Gander and T. Vanzan. Heterogeneous optimized Schwarz methods for second order elliptic PDEs. *SIAM J. Sci. Comp.*, 41(4):A2329–A2354, 2019.
52. M. J. Gander and G. Wanner. From Euler, Ritz, and Galerkin to modern computing. *SIAM Review*, 54(4):627–666, 2012.
53. M. J. Gander and G. Wanner. The origins of the alternating Schwarz method. In *Domain decomposition methods in science and engineering XXI*, pages 487–495. Springer, 2014.
54. M. J. Gander, S.-L. Wu, and T. Zhou. Time parallelization for hyperbolic and parabolic problems. *Acta Numerica*, 2025.

55. M. J. Gander and H. Zhang. A class of iterative solvers for the Helmholtz equation: Factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods. *SIAM Review*, 61(1):3–76, 2019.
56. M. J. Gander and H. Zhang. Schwarz methods by domain truncation. *Acta Numerica*, 31:1–134, 2022.
57. M. H. Gutknecht. A brief introduction to Krylov space methods for solving linear systems. In *Frontiers of Computational Science: Proceedings of the International Symposium on Frontiers of Computational Science 2005*, pages 53–62. Springer, 2007.
58. W. Hackbusch. *Multi-grid methods and applications*, volume 4. Springer, 1985.
59. A. Heinlein, A. Klawonn, J. Knepper, and O. Rheinbach. An adaptive GDSW coarse space for two-level overlapping Schwarz methods in two dimensions. In *Domain Decomposition Methods in Science and Engineering XXIV*, pages 373–382. Springer, 2018.
60. A. Heinlein, A. Klawonn, J. Knepper, and O. Rheinbach. Multiscale coarse spaces for overlapping Schwarz methods based on the ACMS space in 2D. *ETNA*, 48:156–182, 2018.
61. F. Kwok. Is Additive Schwarz with harmonic extension just Lions' method in disguise? In *Domain Decomposition Methods in Science and Engineering XIX*, pages 439–446. Springer, 2011.
62. F. Kwok. Optimized additive Schwarz with harmonic extension as a discretization of the continuous parallel Schwarz method. *SIAM J. Numer. Anal.*, 49(3):1289–1316, 2011.
63. P.-L. Lions. On the Schwarz alternating method. I. In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42, Philadelphia, PA, 1988. SIAM.
64. P.-L. Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1990. SIAM.
65. J. Mandel and M. Brezina. Balancing domain decomposition: Theory and computations in two and three dimensions. Technical Report UCD/CCM 2, Center for Computational Mathematics, University of Colorado at Denver, 1993.
66. F. Nataf, F. Rogier, and E. de Sturler. Domain decomposition methods for fluid dynamics. In A. Sequeira, editor, *Navier-Stokes Equations and Related Nonlinear Analysis*, pages 367–376. Plenum Press Corporation, 1995.
67. F. Nataf, H. Xiang, V. Dolean, and N. Spillane. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM J. Sci. Comp.*, 33(4):1623–1642, 2011.
68. R. A. Nicolaides. Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.*, 24(2):355–365, 1987.
69. A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations.* Oxford University Press, 1999.
70. H. A. Schwarz. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, May 1870.
71. B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, 1996.
72. N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numerische Mathematik*, 126(4):741–770, 2014.

73. C. C. Stolk. A rapidly converging domain decomposition method for the Helmholtz equation. *J. Comp. Phys.*, 241:240–252, 2013.
74. A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, 2005.
75. J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613, 1992.
76. P. Zanolli. Domain decomposition algorithms for spectral approximations. *Calcolo*, pages 201–240, 1988.
77. L. Zepeda-Núñez, R. J. Hewett, and L. Demanet. Preconditioning the 2D Helmholtz equation with polarized traces. In *SEG Technical Program Expanded Abstracts 2014*, pages 3465–3470. SEG, 2014.