



Article scientifique

Article

2003

Accepted version

Open Access

This is an author manuscript post-peer-reviewing (accepted version) of the original publication. The layout of the published version may differ .

---

## Why Restricted Additive Schwarz Converges Faster than Additive Schwarz

---

Efstathiou, Evridiki; Gander, Martin Jakob

### How to cite

EFSTATHIOU, Evridiki, GANDER, Martin Jakob. Why Restricted Additive Schwarz Converges Faster than Additive Schwarz. In: BIT, 2003, vol. 43, n° 5, p. 945–959.

This publication URL: <https://archive-ouverte.unige.ch/unige:6282>

# Why Restricted Additive Schwarz Converges Faster than Additive Schwarz \*

EVRIKIDIKI EFSTATHIOU<sup>1</sup> and MARTIN J. GANDER<sup>1</sup>

<sup>1</sup> *Department of Mathematics and Statistics, McGill University  
Montreal, QC, Canada H3A 2K6. email: efstathiou@math.mcgill.ca,  
mgander@math.mcgill.ca*

## Abstract.

Recently a variant of the additive Schwarz (AS) preconditioner, the restricted additive Schwarz (RAS) preconditioner has been introduced, and numerical experiments showed that RAS converges faster and requires less communication than AS. We show in this paper how RAS, which is defined at the matrix level, can be interpreted as an iteration at the continuous level of the underlying problem. This interpretation reveals why RAS converges faster than classical AS.

*AMS subject classification:* 65F10, 65N22.

*Key words:* Domain Decomposition, Additive Schwarz, Restricted Additive Schwarz, Multiplicative Schwarz, Restricted Multiplicative Schwarz

## 1 Introduction

*The final extension we wish to consider concerns “parallel” versions of the Schwarz alternating method . . . ,  $u_{n+1}^i$  is solution of  $-\Delta u_{n+1}^i = f$  in  $O_i$  and  $u_{n+1}^i = u_n^j$  on  $\partial O_i \cap O_j$ . (Lions, 1988)*

*The basic idea behind the additive form of the algorithm is to work with the simplest possible polynomial in the projections. Therefore the equation  $(P_1 + P_2 + \dots + P_N)u_h = g_h$  is solved by an iterative method. (Dryja and Widlund, 1989)*

The classical alternating Schwarz method was formulated at the continuous level by Schwarz in 1870 to prove existence and uniqueness of solutions of Laplace’s equation on irregular domains [17]. This method was sequential and defined for two subdomains (hence called ‘alternating’ by Schwarz), and thus is like a block Gauss Seidel iteration where the blocks are corresponding to the subdomains. To distinguish this method at the continuous level from the ones defined at the matrix level, we call it in this paper the Gauss Seidel Schwarz (GSS) method. More than a century later the Schwarz method gained popularity with the advent of parallel computers. Lions analyzed the alternating Schwarz method at the continuous level using a projection formulation in [12] and maximum principle arguments in [13] and generalized it to the case of more than two subdomains. He also proposed at the continuous level to do the subdomain solves in parallel, treating the subdomains in a Jacobi fashion, see the quote at the beginning

---

\*Received September 2002. Revised ???. Communicated by Daniel B. Szyld.

of this section. To distinguish this continuous parallel variant of the Schwarz method from the ones defined at the matrix level, we call this method the Jacobi Schwarz (JS) method here.

Dryja and Widlund used the alternating Schwarz method at the continuous level in [6] to introduce the multiplicative Schwarz (MS) preconditioner at the matrix level, which goes sequentially through the subdomains, and is the discrete analog of the continuous GSS method. The projection interpretation of the alternating Schwarz method led the authors also to a more parallel variant, the additive Schwarz (AS) preconditioner defined at the matrix level (see the quote at the beginning of the section). While the AS preconditioner does the subdomain solves simultaneously, in the same spirit as the classical JS method at the continuous level in [12], no link was made between the discrete AS preconditioner and the continuous JS method. This way of introducing the Schwarz preconditioners became classical: first the alternating Schwarz method is introduced at the continuous level, then the discrete MS preconditioner is derived from it, and noticing that it corresponds to a block Gauss Seidel iteration with overlap, the corresponding block Jacobi iteration, the AS preconditioner, is introduced at the discrete level, without a continuous interpretation, see for example the review article by Chan and Mathew [4] or the book by Smith, Bjørstad and Gropp [18]. While in most later papers the condition numbers of the systems preconditioned with MS or AS are studied, the underlying methods are defined as iterative methods, and AS has a particularity, which is not always mentioned in the domain decomposition literature. In [18] we find for example: “In general the AS method will not converge, it is always accelerated with a Krylov subspace method”, and in [11] the AS method is defined with a damping factor  $\Theta$  with the comment “the AS iteration converges for sufficiently small  $\Theta$ ”, see also [8]. The JS method defined at the continuous level however was proved to converge by Lions in [12] without a damping factor.

In a recent paper, Cai and Sarkis introduced a variant of the AS preconditioner called restricted additive Schwarz (RAS), which was found accidentally; we cite from [3]:

While working on an AS/GMRES algorithm in an Euler simulation, we removed part of the communication routine and surprisingly the “then AS” method converged faster in both terms of iteration counts and CPU time.

RAS was further analyzed in [2], [16], [1] and at the algebraic level a convergence theory for RAS is presented in [9]. RAS is now the default preconditioner in the software package PETSc. In their original publication [3] the authors also proposed a restricted multiplicative Schwarz (RMS) preconditioner. A convergence theory for RMS at the algebraic level can be found in [14].

The goal of this paper is to show that RAS, which is formulated at the matrix level, has a continuous interpretation at the level of the differential equation, and it coincides with the JS method that was proposed and analyzed by Lions in [12]. This interpretation allows us to show why RAS has better convergence properties than AS, and hence it complements the convergence theory at the

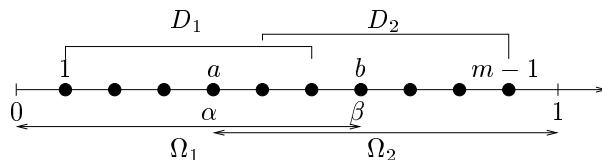


Figure 2.1: Domains and discretization of the Schwarz algorithms.

algebraic level given in [9]. To show this result, we first analyze the model problem of the one dimensional Poisson equation and two subdomains. We then show how the results generalize to higher dimensional problems with more subdomains.

## 2 The Gauss-Seidel and Jacobi Schwarz Methods

To show the main idea, we consider first the Poisson equation in  $\Omega = (0, 1)$ ,

$$u_{xx} = f \quad \text{in } \Omega, \quad u(0) = u(1) = 0.$$

We decompose the domain  $\Omega$  into two overlapping subdomains  $\Omega_1 = (0, \beta)$  and  $\Omega_2 = (\alpha, 1)$  with  $\alpha < \beta$ , as shown in Figure 2.1. The classical alternating Schwarz algorithm (Gauss Seidel Schwarz or GSS) given in [17] is

$$(2.1) \quad \begin{aligned} v_{xx}^n &= f, & \text{on } \Omega_1, & & w_{xx}^n &= f, & \text{on } \Omega_2, \\ v^n(\beta) &= w^{n-1}(\beta), & & & w^n(\alpha) &= v^n(\alpha), \end{aligned}$$

whereas the more parallel Jacobi Schwarz algorithm or JS defined in [12] is given by

$$(2.2) \quad \begin{aligned} v_{xx}^n &= f, & \text{on } \Omega_1, & & w_{xx}^n &= f, & \text{on } \Omega_2, \\ v^n(\beta) &= w^{n-1}(\beta), & & & w^n(\alpha) &= v^{n-1}(\alpha). \end{aligned}$$

In this classical form of the Schwarz algorithms there is no notion of a global approximate solution. To be able to compare these algorithms later with the Schwarz algorithms defined at the matrix level, we note that one could define a global approximation to the solution by choosing the subdomain solution in each subdomain and any weighted average within the overlap,  $u^n = \chi_1 v^n + \chi_2 w^n$  where  $\chi_1 = 1$  in  $\Omega_1 \setminus (\Omega_1 \cap \Omega_2)$ ,  $\chi_2 = 1$  in  $\Omega_2 \setminus (\Omega_1 \cap \Omega_2)$  and  $\chi_1 + \chi_2 = 1$  in the overlap  $\Omega_1 \cap \Omega_2$ . Such a weighting was also used in [20, 5, 19] at the continuous level in the context of parabolic problems and is standard in the context of multi-splittings, see [15].

In an implementation, the continuous Schwarz algorithms are discretized to obtain matrix based Schwarz methods. Choosing for our model problem the standard centered finite difference stencil with  $m - 1$  equally spaced interior nodes  $x_i = ih$ ,  $i = 1, 2, \dots, m - 1$ ,  $h = 1/m$ , we obtain for the discretized global

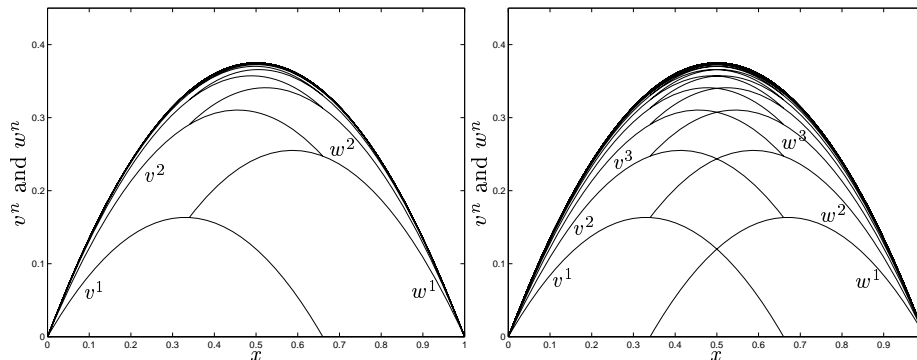


Figure 2.2: Discretizations of the classical Schwarz algorithms, GSS on the left and JS on the right; both are convergent.

problem the linear system

$$A\mathbf{u} = \mathbf{f}, \quad A = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & \ddots \\ & & \ddots & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}, \quad f_i = f(x_i), \quad i = 1, 2, \dots, m-1.$$

Discretizing the algorithms, one finds (see, e.g., [18]), assuming for simplicity that  $\alpha$  and  $\beta$  fall onto the grid-points  $a$  and  $b$  respectively, as shown in Figure 2.1, for the discrete classical GSS method

$$(2.3) \quad A_1 \begin{bmatrix} v_1^n \\ \vdots \\ v_{b-2}^n \\ v_{b-1}^n \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_{b-2} \\ f_{b-1} - \frac{1}{h^2} w_b^{n-1} \end{bmatrix}, \quad A_2 \begin{bmatrix} w_{a+1}^n \\ w_{a+2}^n \\ \vdots \\ w_{m-1}^n \end{bmatrix} = \begin{bmatrix} f_{a+1} - \frac{1}{h^2} v_a^n \\ f_{a+2} \\ \vdots \\ f_{m-1} \end{bmatrix},$$

and for the more parallel JS method

$$(2.4) \quad A_1 \begin{bmatrix} v_1^n \\ \vdots \\ v_{b-2}^n \\ v_{b-1}^n \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_{b-2} \\ f_{b-1} - \frac{1}{h^2} w_b^{n-1} \end{bmatrix}, \quad A_2 \begin{bmatrix} w_{a+1}^n \\ w_{a+2}^n \\ \vdots \\ w_{m-1}^n \end{bmatrix} = \begin{bmatrix} f_{a+1} - \frac{1}{h^2} v_a^{n-1} \\ f_{a+2} \\ \vdots \\ f_{m-1} \end{bmatrix},$$

where  $A_1$  and  $A_2$  are the subdomain matrices of the same form as  $A$  with appropriate size of the discrete subdomains  $D_1$  and  $D_2$ . These algorithms are the precise discrete analogs of the continuous algorithms. A numerical example is shown for both algorithms in Figure 2.2 for  $f = -3$  and  $\Omega_1 = (0, 2/3)$ ,  $\Omega_2 = (1/3, 1)$ . Both algorithms are convergent, and in this simple two subdomain case, the GSS method produces a subsequence of the iterates of the JS method. This is however not true in general, as shown for example in [12].

### 3 The Multiplicative Schwarz Preconditioner

The multiplicative Schwarz preconditioner (MS) at the matrix level is defined by (see, e.g., [4])

$$(3.1) \quad \begin{aligned} \mathbf{u}^{n-\frac{1}{2}} &= \mathbf{u}^{n-1} + R_1^T A_1^{-1} R_1 (\mathbf{f} - A \mathbf{u}^{n-1}), \\ \mathbf{u}^n &= \mathbf{u}^{n-\frac{1}{2}} + R_2^T A_2^{-1} R_2 (\mathbf{f} - A \mathbf{u}^{n-\frac{1}{2}}), \end{aligned}$$

where  $\mathbf{u}^n$  is a global approximation to the solution. Here  $A_1$  and  $A_2$  are the subdomain matrices as before, and  $R_1$  and  $R_2$  are rectangular restriction matrices, the identity on the corresponding subdomains, extended by zeros. Their action restricts a vector in the discrete domain  $D$  to a vector in  $D_1$  and  $D_2$  respectively by choosing the entries corresponding to the interior nodes of the subdomain. Their transpose is an extension matrix that prolongs a vector in  $D_i$  to one in  $D$  by adding zeros for the nodal values of  $D \setminus D_i$ . Using  $R_i$ , one can define the subdomain matrices on  $D_i$  by  $A_i = R_i A R_i^T$ . The MS algorithm (3.1) is identical to its discretized counterpart GSS, a result shown also for more general situations for example in [18] and [4]. Since this identity is non-trivial and we need it later to relate RAS to JS, we show it now for our model problem in detail. For the residual in the  $n - \frac{1}{2}$  step, we get

$$\mathbf{f} - A \mathbf{u}^{n-1} = \begin{bmatrix} f_1 \\ \vdots \\ f_{b-1} \\ f_b \\ \vdots \\ f_{m-1} \end{bmatrix} - \begin{bmatrix} A_1 \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \end{bmatrix} \\ \begin{bmatrix} \frac{1}{h^2} u_b^{n-1} \\ 0 \\ \vdots \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ \frac{1}{h^2} u_b^{n-1} \\ u_b^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix} + B \begin{bmatrix} u_b^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix},$$

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

$$R_1 (\mathbf{f} - A \mathbf{u}^{n-1}) = \begin{bmatrix} f_1 \\ \vdots \\ f_{b-1} - \frac{1}{h^2} u_b^{n-1} \end{bmatrix} - A_1 \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \end{bmatrix}.$$

Next the subdomain solve  $A_1^{-1}$  is applied,

$$(3.2) \quad A_1^{-1} R_1 (\mathbf{f} - A \mathbf{u}^{n-1}) = A_1^{-1} \begin{bmatrix} f_1 \\ \vdots \\ f_{b-1} - \frac{1}{h^2} u_b^{n-1} \end{bmatrix} - \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \end{bmatrix} = \begin{bmatrix} v_1^n \\ \vdots \\ v_{b-1}^n \end{bmatrix} - \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \end{bmatrix}$$

where we introduced the vector  $\mathbf{v}^n$ , since it is precisely a subdomain solve with boundary condition  $u_b^{n-1}$  in GSS, see (2.3). Continuing with the algorithm, we

extend (3.2) to  $D$  and add it to  $\mathbf{u}^{n-1}$  to get

$$\mathbf{u}^{n-\frac{1}{2}} = \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \\ u_b^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix} + A_1^{-1} \begin{bmatrix} f_1 \\ \vdots \\ f_{b-1} - \frac{1}{h^2} u_b^{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \end{bmatrix} = \begin{bmatrix} v_1^n \\ \vdots \\ v_{b-1}^n \\ u_b^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix}.$$

Hence in  $\mathbf{u}^{n-\frac{1}{2}}$ , no matter what the content of the entries  $1 \dots b$  was, we find the solution of a subdomain solve with boundary condition  $u_b^{n-1}$ . For the second part of the algorithm, we find similarly

$$\mathbf{u}^n = \begin{bmatrix} v_1^n \\ \vdots \\ v_a^n \\ v_{a+1}^n \\ \vdots \\ v_{b-1}^n \\ u_b^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix} - A_2^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f_{a+1} - \frac{1}{h^2} v_a^n \\ \vdots \\ f_{b-1} \\ f_b \\ \vdots \\ f_{m-1} \end{bmatrix} - \begin{bmatrix} v_{a+1}^n \\ \vdots \\ v_{b-1}^n \\ u_b^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix} = \begin{bmatrix} v_1^n \\ \vdots \\ v_a^n \\ w_{a+1}^n \\ \vdots \\ w_{b-1}^n \\ w_b^n \\ \vdots \\ w_{m-1}^n \end{bmatrix}$$

where we denote by  $\mathbf{w}^n$  the subdomain solution in  $D_2$  with boundary condition  $v_a^n$ . Comparing with (2.3) we see that this corresponds precisely to the discretized GSS method with the global approximate solution defined by  $\mathbf{u}^n = \chi_1 \mathbf{v}^n + \chi_2 \mathbf{w}^n$  and  $\chi_1 = 1$  in  $D_1 \setminus (D_1 \cap D_2)$ , zero elsewhere, and  $\chi_2 = 1$  in  $D_2$  and zero elsewhere. Hence the MS and the discretized GSS algorithms produce identical iterates, and thus have the same convergence rate.

#### 4 The Additive Schwarz Preconditioner

The additive Schwarz preconditioner (AS) is defined (see, e.g., [4]) by the iteration

$$(4.1) \quad \mathbf{u}^n = \mathbf{u}^{n-1} + (R_1^T A_1^{-1} R_1 + R_2^T A_2^{-1} R_2)(\mathbf{f} - A \mathbf{u}^{n-1}).$$

Tracing each step of this algorithm as before, we find

$$(4.2) \quad \mathbf{u}^n = \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_a^{n-1} \\ u_{a+1}^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \\ u_b^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix} + A_1^{-1} \begin{bmatrix} f_1 \\ \vdots \\ f_a \\ f_{a+1} \\ \vdots \\ f_{b-1} - \frac{1}{h^2} u_b^{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_a^{n-1} \\ u_{a+1}^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \end{bmatrix} + A_2^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f_{a+1} - \frac{1}{h^2} u_a^{n-1} \\ \vdots \\ f_{b-1} \\ f_b \\ \vdots \\ f_{m-1} \end{bmatrix} - \begin{bmatrix} u_{a+1}^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \\ u_b^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix}$$

or with the subdomain solutions  $\mathbf{v}^n$  and  $\mathbf{w}^n$

$$(4.3) \quad \mathbf{u}^n = \begin{bmatrix} v_1^n \\ \vdots \\ v_a^n \\ v_{a+1}^n \\ \vdots \\ v_{b-1}^n \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ w_{a+1}^n \\ \vdots \\ w_{b-1}^n \\ w_b^n \\ \vdots \\ w_{m-1}^n \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_{a+1}^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

which results in the iterative method  $\mathbf{u}^n = M_{AS} \mathbf{u}^{n-1} + \mathbf{b}_{AS}$  where

$$(4.4) \quad M_{AS} = \begin{bmatrix} 0 & & & & & & & & & & & & p_1 \\ & \ddots & & & & & & & & & & & p_2 \\ & & & & & & & & & & & & \vdots \\ & & & 0 & & & & & & & & & \vdots \\ & & & q_1 & -1 & & & & & & & & \vdots \\ & & & q_2 & & \ddots & & & & & & & p_{b-2} \\ & & & \vdots & & & & & & & & & \vdots \\ & & & \vdots & & & -1 & p_{b-1} & & & & & \\ & & & \vdots & & & & 0 & & & & & \\ & & & q_{m-a-2} & & & & & & & \ddots & & \\ & & & q_{m-a-1} & & & & & & & & \ddots & 0 \end{bmatrix}, \quad \mathbf{b}_{AS} = \begin{bmatrix} A_1^{-1} \begin{bmatrix} f_1 \\ \vdots \\ f_a \\ f_{a+1} \\ \vdots \\ f_{b-1} \end{bmatrix} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ A_2^{-1} \begin{bmatrix} f_{a+1} \\ \vdots \\ f_{b-1} \\ f_b \\ \vdots \\ f_{m-1} \end{bmatrix} \end{bmatrix}$$

and the two vectors  $\mathbf{p}$  and  $\mathbf{q}$  we introduced to define  $M_{AS}$  are given by

$$(4.5) \quad \mathbf{p} = -\frac{1}{h^2} A_1^{-1}(:, b-1) \quad \text{and} \quad \mathbf{q} = -\frac{1}{h^2} A_2^{-1}(:, 1).$$

Hence this iterative method does not converge in general, since the spectral radius of the iteration matrix equals one,  $\rho(M) = 1$ . There are as many eigenvalues  $-1$  as nodes in the overlap, for our model problem their number is  $b - a - 1$ , and the corresponding eigenvectors are the unit vectors  $\mathbf{e}_{a+1}, \dots, \mathbf{e}_{b-1}$ . Hence the AS method does not converge in the overlap, only in the interior of the subdomains. This is the particularity of the AS preconditioner, it can only be used either with Krylov acceleration or a damping factor, otherwise the method does not converge. An example is shown in Figure 4.1 on the left, where one can clearly see that the AS method does not converge in the overlap. Hence the AS preconditioner defined in [6] *does not* correspond to the classical, convergent JS algorithm introduced by Lions in [12].



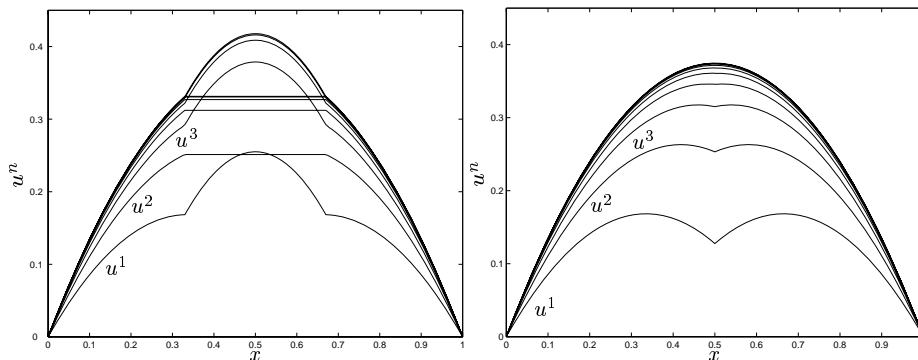


Figure 4.1: Comparison of AS on the left and RAS on the right for the model problem with  $f=-3$ . One can clearly see that AS is not convergent in the overlap, while RAS converges everywhere.

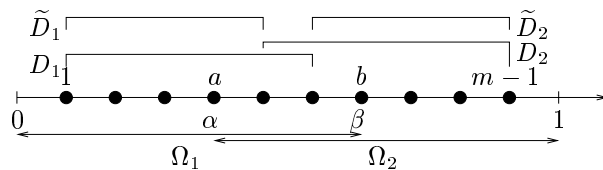


Figure 5.1: The non-overlapping subdomains  $\tilde{D}_i$  for the restriction operators  $\tilde{R}_i$ .

## 5 The Restricted Additive Schwarz Preconditioner

We now show that the RAS preconditioner leads to an iterative method which is identical to the discretization of the continuous JS method introduced in [12] and hence converges without Krylov acceleration or damping factor, which explains its superiority to the AS preconditioner. The RAS preconditioner is defined in [3] by

$$\mathbf{u}^n = \mathbf{u}^{n-1} + (\tilde{R}_1^T A_1^{-1} R_1 + \tilde{R}_2^T A_2^{-1} R_2)(\mathbf{f} - A\mathbf{u}^{n-1}),$$

where  $\tilde{R}_1$  and  $\tilde{R}_2$  are now restriction matrices on two non-overlapping subdomains  $\tilde{D}_1 = \{1 \dots c-1\}$  and  $\tilde{D}_2 = \{c \dots m-1\}$  respectively with  $a < c \leq b$ , as shown in Figure 5.1. To see why RAS corrects the problem of AS in the overlap, we only need to replace the extension operators in what we found for AS in (4.2),

$$\mathbf{u}^n = \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_c^{n-1} \\ \vdots \\ u_{m-2}^{n-1} \\ u_{m-1}^{n-1} \end{bmatrix} + \tilde{R}_1^T A_1^{-1} \begin{bmatrix} f_1 \\ \vdots \\ f_c \\ \vdots \\ f_{b-2} \\ f_{b-1} - \frac{1}{h^2} u_b^{n-1} \end{bmatrix} - \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_c^{n-1} \\ \vdots \\ u_{b-2}^{n-1} \\ u_{b-1}^{n-1} \end{bmatrix} + \tilde{R}_2^T A_2^{-1} \begin{bmatrix} f_{a+1} - \frac{1}{h^2} u_a^{n-1} \\ f_{a+2} \\ \vdots \\ f_c \\ \vdots \\ f_{m-1} \end{bmatrix} - \begin{bmatrix} u_{a+1}^{n-1} \\ u_{a+2}^{n-1} \\ \vdots \\ u_c^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix}.$$

Using again  $\mathbf{v}^n$  and  $\mathbf{w}^n$  to denote the subdomain solves with boundary condition  $u_b^{n-1}$  and  $u_a^{n-1}$  respectively, we find now with these new extension operators

$$\mathbf{u}^n = \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{c-1}^{n-1} \\ u_c^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix} + \begin{bmatrix} v_1^n \\ \vdots \\ v_{c-1}^n \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{c-1}^{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ w_c^n \\ \vdots \\ w_{m-1}^n \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_c^{n-1} \\ \vdots \\ u_{m-1}^{n-1} \end{bmatrix} = \begin{bmatrix} v_1^n \\ \vdots \\ v_{c-1}^n \\ w_c^n \\ \vdots \\ w_{m-1}^n \end{bmatrix}.$$

Hence RAS corresponds to the discretization of the continuous JS method, with the choice of defining the global approximate solution by  $\mathbf{u}^n = \chi_1 \mathbf{v}^n + \chi_2 \mathbf{w}^n$  and  $\chi_1 = 1$  in  $\tilde{D}_1$ , zero elsewhere, and  $\chi_2 = 1$  in  $\tilde{D}_2$  and zero elsewhere. Hence RAS and the discretized JS algorithms are producing identical iterates and thus have the same convergence rate. A numerical example is shown in Figure 4.1 on the right.

Since AS is not convergent as an iterative method, we need to either use Krylov acceleration or a damping factor to be able to further compare the performance of the two methods. Writing RAS in the form  $\mathbf{u}^n = M_{RAS} \mathbf{u}^{n-1} + \mathbf{b}_{RAS}$ , as we did for AS, we find

$$M_{RAS} = \begin{bmatrix} 0 & & & & p_1 \\ & \ddots & & & p_2 \\ & & 0 & & \vdots \\ & & \vdots & & p_{c-1} \\ & q_{c-a} & \ddots & & 0 \\ & \vdots & & & \vdots \\ & \vdots & & & 0 \\ & q_{m-a-2} & & \ddots & \\ q_{m-a-1} & & & & 0 \end{bmatrix}, \quad \mathbf{b}_{RAS} = \tilde{R}_1^T A_1^{-1} \begin{bmatrix} f_1 \\ \vdots \\ f_a \\ f_{a+1} \\ \vdots \\ f_{b-1} \end{bmatrix} + \tilde{R}_2^T A_2^{-1} \begin{bmatrix} f_{a+1} \\ \vdots \\ f_{b-1} \\ f_b \\ \vdots \\ f_{m-1} \end{bmatrix}$$

with the vectors  $\mathbf{p}$  and  $\mathbf{q}$  as defined in (4.5), but note that now only the entries  $\mathbf{p}(1 : c-1)$  and  $\mathbf{q}(c-a : m-a-1)$  are used in the definition of  $M_{RAS}$ . Using an idea from [10], this shows that RAS used as a preconditioner for a Krylov method converges in at most 3 iterations for this model problem, since  $M_{RAS}$  is a matrix of rank 2, which implies that the Krylov space corresponding to  $I - M_{RAS}$  can have at most dimension 3. On the other hand,  $M_{AS}$  has four distinct eigenvalues, one more than RAS, because of the additional  $-1$  eigenvalue due to the problem of AS in the overlap, and thus a Krylov method preconditioned with AS will converge in 4 iterations, one more than RAS, for this one dimensional example. This is also observed in our numerical experiments: when GMRES is preconditioned with AS for our model problem, it converges in 4 iterations to a tolerance of  $10^{-6}$ , whereas RAS only takes 3 iterations for the same tolerance.

To compare RAS and AS as iterative methods, we introduced the damping factor  $\gamma$  in the iteration for AS,

$$\mathbf{u}^n = \gamma(M_{AS}\mathbf{u}^{n-1} + \mathbf{b}_{AS}) + (1 - \gamma)\mathbf{u}^{n-1}.$$

The optimal damping factor for AS for the one dimensional problem considered here was found to be  $\gamma = 0.8235$ . Using this value, AS converged in 31 steps to the tolerance of  $10^{-6}$ , whereas RAS without damping factor converged in 24 steps to the same tolerance. The damping factor shifts the  $-1$  eigenvalue of the non-convergent eigenmodes of AS into the interval  $(-1, 1)$ , but at the cost of shifting other eigenvalues closer to the boundary of this interval, making the method slower than RAS. To make the comparison even fairer in general, one should also allow a damping factor for the RAS iteration, but for our model problem this was not necessary, since the optimal damping factor for RAS turned out to be one.

## 6 The Restricted Multiplicative Schwarz Preconditioner

We have seen already in Section 3 that the MS preconditioner and the discretization of GSS are identical. We now analyze the RMS preconditioner, defined in [3] by

$$(6.1) \quad \begin{aligned} \mathbf{u}^{n-\frac{1}{2}} &= \mathbf{u}^{n-1} + \tilde{R}_1^T A_1^{-1} R_1 (\mathbf{f} - A\mathbf{u}^{n-1}), \\ \mathbf{u}^n &= \mathbf{u}^{n-\frac{1}{2}} + \tilde{R}_2^T A_2^{-1} R_2 (\mathbf{f} - A\mathbf{u}^{n-\frac{1}{2}}). \end{aligned}$$

Starting from equation (3.2), up to which MS and RMS are identical, we now have to apply  $\tilde{R}_1^T$ ,

$$\tilde{R}_1^T A_1^{-1} R_1 (\mathbf{f} - A\mathbf{u}^{n-1}) = \tilde{R}_1^T \left( \begin{bmatrix} v_1^n \\ \vdots \\ v_{c-1}^n \\ v_c^n \\ \vdots \\ v_{b-1}^n \end{bmatrix} - \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{c-1}^{n-1} \\ u_c^{n-1} \\ \vdots \\ u_{b-1}^{n-1} \end{bmatrix} \right) = \begin{bmatrix} v_1^n \\ \vdots \\ v_{c-1}^n \\ 0 \\ \vdots \\ 0 \end{bmatrix} - \begin{bmatrix} u_1^{n-1} \\ \vdots \\ u_{c-1}^{n-1} \end{bmatrix}.$$

Hence we get the half-step approximation  $\mathbf{u}^{n-\frac{1}{2}} = [v_1^n, \dots, v_{c-1}^n, u_c^{n-1}, \dots, u_{m-1}^{n-1}]^T$  and similarly the full step approximation  $\mathbf{u}^n = [v_1^n, \dots, v_{c-1}^n, w_c^n, \dots, w_{m-1}^n]^T$ . Hence RMS corresponds to the classical GSS method with the global approximate solution defined as in RAS. Note that this way of defining the global solution can lead to discontinuities of the approximate solution in the overlap; a comparison of MS and RMS is shown in Figure 6.1.

## 7 Generalization

The results shown above for the Poisson equation in one dimension and two subdomains hold also in a more general setting, only the notation becomes more

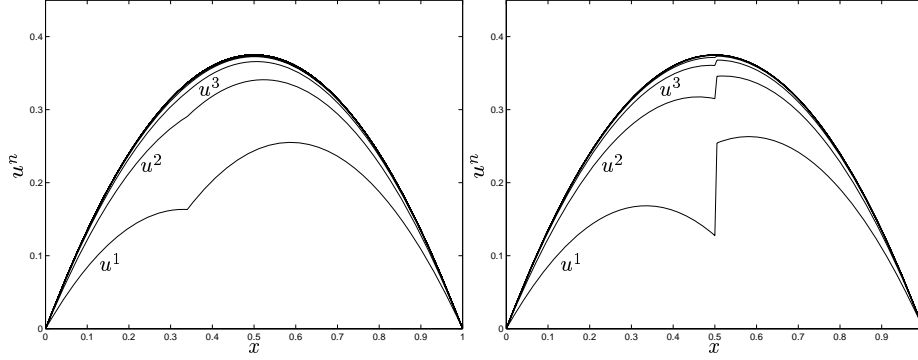


Figure 6.1: MS on the left and RMS on the right for our model problem with  $f=-3$ . Both methods are convergent at the same rate, but the iterates of RMS can form discontinuities.

involved. To get the idea, we show the result for AS only for the two dimensional Poisson equation

$$(7.1) \quad u_{xx} + u_{yy} = f \quad \text{on } \Omega, \quad u = 0 \quad \text{on } \partial\Omega,$$

and the three subdomains  $\Omega_1 = [0, \beta_1] \times [0, 1]$ ,  $\Omega_2 = [\alpha_1, \beta_2] \times [0, 1]$ ,  $\Omega_3 = [\alpha_2, 2] \times [0, 1]$ , as shown in Figure 7.1. The AS iteration in that case is given by

$$(7.2) \quad \mathbf{u}^n = \mathbf{u}^{n-1} + (R_1^T A_1^{-1} R_1 + R_2^T A_2^{-1} R_2 + R_3^T A_3^{-1} R_3)(\mathbf{f} - A\mathbf{u}^{n-1}),$$

where  $A$  now represents the standard five point finite difference discretization of the Laplacian in two dimensions and the  $R_j$ ,  $j = 1, 2, 3$ , are the restriction matrices onto the nodes of the discretized subdomains  $D_j$ ,  $j = 1, 2, 3$ , as shown in Figure 7.1. Without loss of generality we assume equal grid spacing  $h$  in the  $x$  and  $y$  direction to keep the notation simpler. We denote for any grid function  $\mathbf{f}$  approximating a continuous function  $f(x, y)$  by  $f_{i,j} \approx f(ih, jh)$ .

As in the one dimensional case with two subdomains treated in Section 4, we trace the main steps of the algorithm (7.2) in detail. The local solve applied to the residual  $\mathbf{f} - A\mathbf{u}^{n-1}$  on each subdomain gives for the first subdomain

$$A_1^{-1} R_1(\mathbf{f} - A\mathbf{u}^{n-1}) = A_1^{-1} \begin{bmatrix} f_{1,1} \\ \vdots \\ f_{b_1-2,m-1} \\ f_{b_1-1,1} - \frac{1}{h^2} u_{b_1,1}^{n-1} \\ \vdots \\ f_{b_1-1,m-1} - \frac{1}{h^2} u_{b_1,m-1}^{n-1} \end{bmatrix} - \begin{bmatrix} u_{1,1}^{n-1} \\ \vdots \\ u_{b_1-2,m-1}^{n-1} \\ u_{b_1-1,1}^{n-1} \\ \vdots \\ u_{b_1-1,m-1}^{n-1} \end{bmatrix},$$

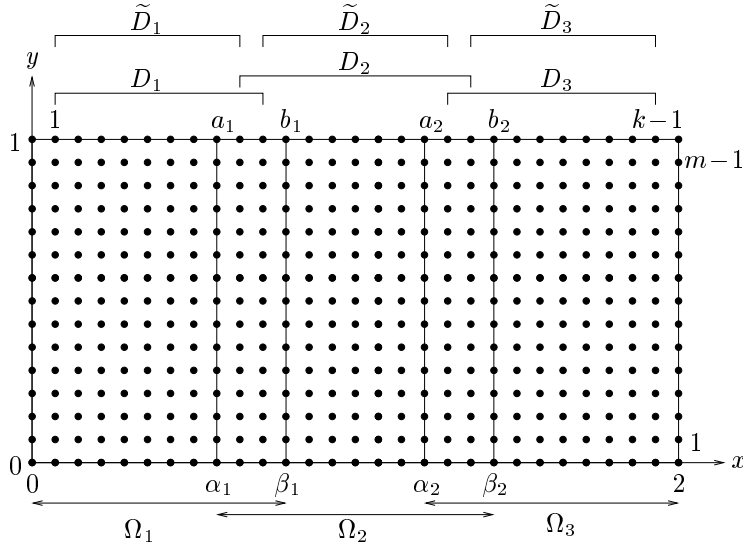


Figure 7.1: An example in two dimensions with three subdomains.

on the second subdomain

$$A_2^{-1}R_2(\mathbf{f} - A\mathbf{u}^{n-1}) = A_2^{-1} \begin{bmatrix} f_{a_1+1,1} - \frac{1}{h^2}u_{a_1,1}^{n-1} \\ \vdots \\ f_{a_1+1,m-1} - \frac{1}{h^2}u_{a_1,m-1}^{n-1} \\ f_{a_1+2,1} \\ \vdots \\ f_{b_2-2,m-1} \\ f_{b_2-1,1} - \frac{1}{h^2}u_{b_2,1}^{n-1} \\ \vdots \\ f_{b_2-1,m-1} - \frac{1}{h^2}u_{b_2,m-1}^{n-1} \end{bmatrix} = \begin{bmatrix} u_{a_1+1,1}^{n-1} \\ \vdots \\ u_{a_1+1,m-1}^{n-1} \\ u_{a_1+2,1}^{n-1} \\ \vdots \\ u_{b_2-2,m-1}^{n-1} \\ u_{b_2-1,1}^{n-1} \\ \vdots \\ u_{b_2-1,m-1}^{n-1} \end{bmatrix},$$

and on the last subdomain

$$A_3^{-1}R_3(\mathbf{f} - A\mathbf{u}^{n-1}) = A_3^{-1} \begin{bmatrix} f_{a_2+1,1} - \frac{1}{h^2}u_{a_2,1}^{n-1} \\ \vdots \\ f_{a_2+1,m-1} - \frac{1}{h^2}u_{a_2,m-1}^{n-1} \\ f_{a_2+2,1} \\ \vdots \\ f_{k-1,m-1} \end{bmatrix} = \begin{bmatrix} u_{a_2+1,1}^{n-1} \\ \vdots \\ u_{a_2+1,m-1}^{n-1} \\ u_{a_2+2,1}^{n-1} \\ \vdots \\ u_{k-1,m-1}^{n-1} \end{bmatrix}.$$

Denoting the local solves as before by  $\mathbf{v}^n$ ,  $\mathbf{w}^n$  and  $\mathbf{z}^n$  respectively, extending the results to the entire domain using the extension matrices  $R_j^T$ ,  $j = 1, 2, 3$  and

adding the old iterate  $\mathbf{u}^{n-1}$  according to the algorithm (7.2), we obtain the new iterate

$$(7.3) \quad \mathbf{u}^n = \begin{bmatrix} v_{1,1}^n \\ \vdots \\ v_{a_1,m-1}^n \\ v_{a_1+1,1}^n \\ \vdots \\ v_{b_1-1,m-1}^n \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ w_{a_1+1,1}^n \\ \vdots \\ w_{b_1-1,m-1}^n \\ w_{b_1,1}^n \\ \vdots \\ w_{a_2,m-1}^n \\ w_{a_2+1,1}^n \\ \vdots \\ w_{b_2-1,m-1}^n \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ z_{a_2+1,1}^n \\ \vdots \\ z_{b_2-1,m-1}^n \\ z_{b_2,1}^n \\ \vdots \\ z_{k-1,m-1}^n \end{bmatrix} - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_{a_1+1,1}^{n-1} \\ \vdots \\ u_{b_1-1,1}^{n-1} \\ 0 \\ \vdots \\ 0 \\ u_{a_2+1,1}^{n-1} \\ \vdots \\ u_{b_2-1,1}^{n-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

which shows that as in the simpler one dimensional case with two subdomains given in (4.3), the iteration matrix here also has a spectral radius equal to one and AS does not converge in the overlap. Using however instead of  $R_j^T$  in the extension process  $\tilde{R}_j^T$  associated with the non-overlapping decomposition  $\tilde{D}_j$ , as it is done in RAS, the last vector on the right hand side in (7.3) depending on  $\mathbf{u}^{n-1}$  disappears and the subdomain solutions are put together without overlap. Hence the method becomes the discretized classical JS algorithm defined by Lions in [12] also in this higher dimensional case with 3 subdomains.

## 8 Conclusions

Using a continuous interpretation of the RAS preconditioner we have shown why RAS has better convergence properties than AS. It is due to the fact that, when used as iterative solvers, RAS is convergent everywhere, whereas AS is not convergent in the overlap. Away from the overlap, the iterates are identical. This observation holds not only for discretized partial differential equations, it is true for arbitrary matrix problems, see [7]. Nevertheless, the AS preconditioner can and has been successfully used, either together with Krylov acceleration, or with a damping factor. But we have shown that the non-convergent modes stemming from the problem in the overlap of AS make the method slower than RAS in both those cases. For the important class of symmetric problems, RAS has the disadvantage of being non-symmetric, and hence a Krylov method for non-symmetric problems needs to be used, whereas AS for symmetric problems is symmetric. It is to our knowledge an open problem if JS can be written in symmetric form for symmetric problems.

We close this paper with a remark on the communication: while the discretizations of the continuous GSS and JS methods need to communicate interface values only, the MS, AS, RAS and RMS algorithm formulations require the exchange of information in the entire overlap, which is an order of magnitude more than the information at the interfaces. It might therefore be beneficial for implementations to use the discretized GSS and JS formulations which have identical convergence rates, but with less communication overhead.

#### REFERENCES

1. X. C. Cai, M. Dryja, and M. Sarkis. A restricted additive Schwarz preconditioner with harmonic overlap for symmetric positive definite linear systems. *SIAM J. Sci. Comp.*, 2002. Submitted.
2. X.-C. Cai, C. Farhat, and M. Sarkis. A minimum overlap restricted additive Schwarz preconditioner and applications to 3D flow simulations. *Contemporary Mathematics*, 218:479–485, 1998.
3. X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21:239–247, 1999.
4. T. F. Chan and T. P. Mathew. Domain decomposition algorithms. In *Acta Numerica 1994*, pages 61–143. Cambridge University Press, 1994.
5. H. K. Dahle, T. O. W. Johansen, T. Botnen, and X.-C. Tai. A characteristic domain decomposition method for modeling flow in a coastal aquifer. In P. E. Bjørstad, M. S. Espedal, and D. E. Keyes, editors, *Ninth International Conference on Domain Decomposition Methods*, pages 698–707. ddm.org, 1998.
6. M. Dryja and O. B. Widlund. Some domain decomposition algorithms for elliptic problems. In L. Hayes and D. Kincaid, editors, *Iterative Methods for Large Linear Systems*, pages 273–291, San Diego, California, 1989. Academic Press. Proceeding of the Conference on Iterative Methods for Large Linear Systems held in Austin, Texas, October 19 - 21, 1988, to celebrate the sixty-fifth birthday of David M. Young, Jr.
7. E. Efstathiou. Study of the schwarz algorithms: Understanding restricted additive schwarz. Master’s thesis, McGill University, 2003.
8. A. Frommer and D. B. Szyld. Weighted max norms, splittings, and overlapping additive Schwarz iterations. *Numerische Mathematik*, 83:259–278, 1999.
9. A. Frommer and D. B. Szyld. An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM Journal on Numerical Analysis*, 39:463–479, 2001.
10. G. H. Golub. Convergence Property of Discretized Domain Decomposition Methods. *Personal communication*, 2001.

11. W. Hackbusch. *Iterative Solution of Large Sparse Linear Systems of Equations*. Springer-Verlag, Berlin, 1994.
12. 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.
13. P.-L. Lions. On the Schwarz alternating method. II. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Domain Decomposition Methods*, pages 47–70, Philadelphia, PA, 1989. SIAM.
14. R. Nabben and D. B. Szyld. Convergence theory of restricted multiplicative Schwarz methods. Technical Report 01-5-17, Temple University, May 2001.
15. D. P. O’Leary and R. E. White. Multi-splittings of matrices and parallel solution of linear systems. *SIAM J. Alg. Disc. Meth.*, 6:630–640, 1985.
16. M. Sarkis and B. Koobus. A scaled and minimum overlap restricted additive Schwarz method with application on aerodynamics. *Comp. Meth. Appl. Mech. and Eng.*, 184:391–400, 2000.
17. H. A. Schwarz. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, May 1870.
18. B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.
19. X.-C. Tai. A space decomposition method for parabolic equations. *Numer. Method for Part. Diff. Equat.*, 14(1):27–46, 1998.
20. X.-C. Tai, T. O. W. Johansen, H. K. Dahle, and M. S. Espedal. A characteristic domain splitting method. In R. Glowinski, J. Périaux, Z.-C. Shi, and O. Widlund, editors, *Domain Decomposition Methods in Science and Engineering*, pages 317–323. John Wiley & Sons Ltd., 1996.