Proceedings of the Twelfth International Conference on
Engineering Computational Technology
Edited by: P. Iványi, J. Kruis and B.H.V. Topping
Civil-Comp Conferences, Volume 8, Paper 5.3
Civil-Comp Press, Edinburgh, United Kingdom, 2024
ISSN: 2753-3239, doi: 10.4203/ccc.8.5.3

Hybrid Synchronous-Asynchronous Parallel Computing

G. Gbikpi-Benissan[1] and F. Magoulès[1,2]

[1] **CentraleSupélec, Paris-Saclay University, Gif-sur-Yvette, France**
[2] **Faculty of Engineering and Information Technology, University of Pécs, Hungary**

Abstract

From the very beginning of parallel numerical computing, its major difficulty consists in achieving highly scaling solvers. This is due to its inherent synchronization phases, which are required after each iteration in traditional iterative computing. Asynchronous iterative computing early arose to overcome such a restriction, which makes it an attractive approach towards high performance computing. Nevertheless, asynchronous convergence is guaranteed so far for relaxation methods only, which do not provide the highest convergence rates in many situations. Efforts to achieve more competitive asynchronous solvers include their design within domain decomposition frameworks and, recently, with coarse-grid correction. We propose here a new formulation of asynchronous additive coarse-grid correction, which allows for convergence conditions independent of communications delays. Our approach is based on a two-splitting formulation of the two-level solver, leading to a hybrid synchronous-asynchronous method where asynchronous iterations are combined in parallel with synchronous ones. This possibly paves the way to asynchronous convergence acceleration using quite effective relaxation coefficients obtained from synchronized data.

Keywords: parallel computing, domain decomposition methods, Schwarz-type methods, asynchronous iterations, coarse-grid correction, non-blocking synchronization.

1 Introduction

We address systems of $n \in \mathbb{N}$ linear algebraic equations,

$$Ax = b, \qquad b \in \mathbb{R}^n, \tag{1}$$

where A is a nonsingular real coefficients matrix. Our context is such that A is highly sparse and large, which makes it more efficient to solve the problem (1) using a parallel iterative method. One of the main challenges in parallel iterative computing is to design highly scaling solvers, which is made difficult by the need for the processes to synchronize with each other after each iteration. Asynchronous iterative computing was early introduced to mitigate the issue (see [1, 2]), however, their convergence theory is limited to contraction properties of relaxation methods (see, e.g., [2–6]). Efforts are therefore made to achieve fast-converging asynchronous methods, similarly to the efforts in accelerating classical relaxation methods, for instance, for fluid-structure simulations (see, e.g., [7]).

Our interest here is the extension of algebraic multigrid techniques [8] to asynchronous domain decomposition methods (DDM) [9, 10]. Recent notable results include asynchronous coarse-grid correction schemes developed within domain decomposition frameworks of either additive Schwarz type (see [11–13]) or primal substructuring type (see [14]). We shall consider an additive Schwarz-type domain decomposition framework that covers the augmented block Jacobi representation [15] of the parallel Schwarz method by Lions [16], as well as both the restricted and the weighted additive Schwarz methods by Cai and Sarkis [17]. Extension to primal substructuring frameworks [18] can be derived as in [14]. An overview of Schwarz-type methods can be found in [19]. For a comprehensive introduction to DDM, the reader is referred to [20, 21].

As in [11, 12, 22], we shall consider the additive coarse-grid correction case. The main drawback of those current asynchronous coarse-grid correction schemes is that the convergence of the two-level asynchronous solver requires a limited use of the coarse global information, depending on communications delays. We therefore introduce here a new formulation not requiring any special restriction on the coarse grid incorporation. In the next sections, we give the precise Schwarz-type DDM framework and recall the classical additive coarse-grid correction scheme. Then, we present the new asynchronous formulation with convergence analysis. Some numerical scaling results are discussed, and we conclude with some new perspectives enabled by our approach.

2 Background

Considering the problem (1), let $\mathcal{G}(A)$ denote the adjacency graph of A, where the vertices exactly match the unknowns, and the nonzero coefficients result in the edges.

From a partitioning of $\mathcal{G}(A)$ (see, e.g., [23]), we consider $p \in \mathbb{N}$ disjoint sets of vertices, which can be made overlapping by recursively adding, to each set, all the vertices at a one-edge distance. Let R_i with $i \in \{1, \ldots, p\}$ denote the rectangular matrix of 0 and 1 such that $x_i = R_i x$ correspond to the vector of the unknowns on the i-th set. We define the associated matrix and right-hand-side (RHS) vector,

$$A_i = R_i A R_i^{\mathsf{T}}, \quad b_i = R_i b, \qquad i \in \{1, \ldots, p\}. \tag{2}$$

Note that the matrix R_i is for the mathematical formulation only, and is not explicitly built in practice. Instead, A_i and b_i are directly obtained from selected rows and columns of A and b. We consider Schwarz-type preconditioners of the form

$$M = \sum_{i=1}^{p} R_i^{\mathsf{T}} W_i A_i^{-1} R_i, \tag{3}$$

where W_i, $i \in \{1, \ldots, p\}$, is a diagonal matrix such that

$$\sum_{i=1}^{p} R_i^{\mathsf{T}} W_i R_i = I \tag{4}$$

with I denoting the identity matrix.

Now, as in [20], let R_0 denote the rectangular aggregation matrix giving a coarse version of A as

$$A_0 = R_0 A R_0^{\mathsf{T}}. \tag{5}$$

Each row in R_0 contains the coefficients for aggregating the entries of a given vector to form the corresponding entry of the resulting coarse vector. The coefficients are usually such that the rows in R_0 form a partition of unity. It is common in DDM to aggregate values on each entire vertices set (or its initial nonoverlapping version) into one coarse value. Additive coarse-grid correction is then given by a preconditioner of the form $1/2\left(M + R_0^{\mathsf{T}} A_0^{-1} R_0\right)$.

As mentioned above in introduction, we are interested in deriving an asynchronous variant of two-level Schwarz-type iterative solvers given by

$$x^{k+1} = x^k + \frac{1}{2}\left(M + R_0^{\mathsf{T}} A_0^{-1} R_0\right)\left(b - A x^k\right), \quad k \in \mathbb{N}, \tag{6}$$

where k is the iteration number.

3 Method

The starting point of our approach consists in noticing that the iterative model (6) can actually take the two-splitting form

$$x^{k+1} = \frac{1}{2}\left(x^k + M\left(b - Ax^k\right)\right) + \frac{1}{2}\left(x^k + R_0^{\mathsf{T}} A_0^{-1} R_0\left(b - Ax^k\right)\right), \quad k \in \mathbb{N}, \tag{7}$$

which exhibits the additive combination of one relaxation based on M, and another relaxation based on $R_0^{\mathsf{T}} A_0^{-1} R_0$. Our idea is to consider a hybrid synchronous-asynchronous approach where the M-based relaxation,

$$y^k := x^k + M\left(b - Ax^k\right), \tag{8}$$

will be asynchronous, the $R_0^{\mathsf{T}} A_0^{-1} R_0$-based relaxation,

$$z^k := x^k + R_0^{\mathsf{T}} A_0^{-1} R_0\left(b - Ax^k\right), \tag{9}$$

will remain synchronous, and their weighted sum,

$$x^{k+1} = \frac{1}{2} y^k + \frac{1}{2} z^k, \tag{10}$$

will also be asynchronous.

Using the preconditioner form (3) and the partition of unity (4), the relaxation (8) can take the parallel form

$$y^k := \sum_{i=1}^{p} R_i^{\mathsf{T}} W_i \left(R_i x^k + A_i^{-1} R_i\left(b - Ax^k\right)\right),$$

which restriction to each vertices set results in

$$y_i^k := R_i \sum_{j=1}^{p} R_j^{\mathsf{T}} W_j \left(R_j x^k + A_j^{-1} R_j\left(b - Ax^k\right)\right), \quad i \in \{1, \ldots, p\}. \tag{11}$$

Note that $y_i^k = R_i y^k$. In asynchronous iterative computing, the solution on each vertices set is updated without necessarily synchronizing with the other sets, therefore, there is no global iteration in practice. For theoretical purposes, an iteration will consist in the update of the solution on any vertices set. Let then $\mathcal{S}_k \subset \{1, \ldots, p\}$ give the vertices sets on which the solution vector is concurrently updated at the iteration $k+1$. Due to communications delays, the relaxation is potentially based on outdated versions of the solution vector, instead of its iteration k version. Let $\tau_{i,j}$ with $i, j \in \{1, \ldots, p\}$ denote a function of k such that $\tau_{i,j}(k)$ correspond to the iteration number of the version of the solution vector on the set j from which the set i is updated at the iteration $k+1$. The asynchronous formulation of the relaxation (11) is then given by

$$y_i^k := R_i \sum_{j=1}^{p} R_j^{\mathsf{T}} W_j \left(R_j x^{\tau_{i,j}(k)} + A_j^{-1} R_j\left(b - Ax^{\tau_{i,j}(k)}\right)\right), \qquad i \in \mathcal{S}_k. \tag{12}$$

Similarly, the parallel formulation of the relaxation (9) usually takes the form

$$z_i^k := R_i x^k + R_i R_0^\mathsf{T} A_0^{-1} R_0 \sum_{j=1}^{p} R_j^\mathsf{T} W_j R_j \left(b - A x^k\right), \quad i \in \{1, \ldots, p\}, \tag{13}$$

which exhibits two communications points through $R_0 R_j^\mathsf{T}$ and $R_i R_0^\mathsf{T}$, respectively. Based on the asynchronous model (12), communications delays therefore imply

$$z_i^k := R_i x^{\tau_{i,i}(k)} + R_i R_0^\mathsf{T} A_0^{-1} R_0 \sum_{j=1}^{p} R_j^\mathsf{T} W_j R_j \left(b - A x^{\tau_{0,j}(\tau_{i,0}(k))}\right), \quad i \in \mathcal{S}_k,$$

where the functions $\tau_{0,j}$ and $\tau_{i,0}$ are associated to the two communications points $R_0 R_j^\mathsf{T}$ and $R_i R_0^\mathsf{T}$ mentioned above. To perform the synchronous relaxation (13) in the globally asynchronous context, we consider a delay function, say τ_0, that is independent of the variables i and j. This results in

$$z_i^k := R_i x^{\tau_0(k)} + R_i R_0^\mathsf{T} A_0^{-1} R_0 \sum_{j=1}^{p} R_j^\mathsf{T} W_j R_j \left(b - A x^{\tau_0(k)}\right), \quad i \in \mathcal{S}_k. \tag{14}$$

Finally, the restriction of the sum (10) to each vertices set takes the asynchronous parallel form

$$x_i^{k+1} = \begin{cases} \dfrac{1}{2} y_i^{\tau_{i,i}(k)} + \dfrac{1}{2} z_i^{\tau_{i,i}(k)}, & i \in \mathcal{S}_k, \\ x_i^k, & i \in \{1, \ldots, p\} \setminus \mathcal{S}_k. \end{cases} \tag{15}$$

For convergence analysis, we shall highlight three natural assumptions. First, access to data on the i-th vertices set, $i \in \{1, \ldots, p\}$, for updating the i-th vertices set is instantaneous. Second, no set definitively stops being updated (before global convergence is notified), and third, no set definitively stops being accessed for updated data, for updating any other set. Those assumptions can be formulated as

$$\tau_{i,i}(k) = k, \quad |\{k : i \in \mathcal{S}_k\}| = \infty, \quad \lim_{k\to\infty} \tau_{i,j}(k) = \infty, \quad \lim_{k\to\infty} \tau_0(k) = \infty. \tag{16}$$

Let an M-matrix denote a nonsingular matrix with no positive off-diagonal entry, and which inverse is nonnegative. We have the following convergence result similar to [12] but not requiring to alternate between one-level and two-level iterations.

Theorem 1. *Under the assumptions* (16)*, the two-level Schwarz-type asynchronous iterations* (12)*,* (14)*,* (15) *converge to the solution of the problem* (1) *if A is an* M*-matrix, A_0 is an* M*-matrix, and $I - R_0^\mathsf{T} A_0^{-1} R_0 A \geq 0$.*

Proof. Using the partition of unity (4) and the assumption that $\tau_{i,i}(k) = k$, the model (12), (14), (15) results, for $i \in \mathcal{S}_k$, in

$$\begin{aligned} x_i^{k+1} &= \frac{1}{2} R_i \sum_{j=1}^{p} R_j^{\mathsf{T}} W_j \left(R_j x^{\tau_{i,j}(k)} + A_j^{-1} R_j \left(b - A x^{\tau_{i,j}(k)} \right) \right) \\ &\quad + \frac{1}{2} R_i \left(x^{\tau_0(k)} + R_0^{\mathsf{T}} A_0^{-1} R_0 \left(b - A x^{\tau_0(k)} \right) \right) \\ &= \frac{1}{2} R_i \sum_{j=1}^{p} R_j^{\mathsf{T}} W_j R_j \left(x^{\tau_{i,j}(k)} + R_j^{\mathsf{T}} A_j^{-1} R_j \left(b - A x^{\tau_{i,j}(k)} \right) \right) \\ &\quad + \frac{1}{2} R_i \left(x^{\tau_0(k)} + R_0^{\mathsf{T}} A_0^{-1} R_0 \left(b - A x^{\tau_0(k)} \right) \right). \end{aligned}$$

Following, e.g., [24], matrices M_i^{-1} with $i \in \{1, \ldots, p\}$ can be derived from $R_i^{\mathsf{T}} A_i^{-1} R_i$ such that

$$M = \sum_{i=1}^{p} E_i M_i^{-1}, \quad E_i = R_i^{\mathsf{T}} W_i R_i. \tag{17}$$

Then, we equivalently have

$$\begin{aligned} x_i^{k+1} &= \frac{1}{2} R_i \sum_{j=1}^{p} E_j \left(x^{\tau_{i,j}(k)} + M_j^{-1} \left(b - A x^{\tau_{i,j}(k)} \right) \right) \\ &\quad + \frac{1}{2} R_i \left(x^{\tau_0(k)} + R_0^{\mathsf{T}} A_0^{-1} R_0 \left(b - A x^{\tau_0(k)} \right) \right) \\ &= R_i F \left(x^{\tau_{i,1}(k)}, \ldots, x^{\tau_{i,p}(k)}, x^{\tau_0(k)} \right) \end{aligned}$$

with, for any $X_1, \ldots, X_p, X_0 \in \mathbb{R}^n$,

$$\begin{aligned} F\left(X_1, \ldots, X_p, X_0\right) &= \frac{1}{2} \left(F_1\left(X_1, \ldots, X_p\right) + F_2\left(X_0\right) \right), \\ F_1\left(X_1, \ldots, X_p\right) &= \sum_{i=1}^{p} E_i \left(X_i + M_i^{-1} \left(b - A X_i \right) \right), \\ F_2\left(X_0\right) &= X_0 + R_0^{\mathsf{T}} A_0^{-1} R_0 \left(b - A X_0 \right). \end{aligned}$$

It follows that, for any $X = \left(X_1, \ldots, X_p, X_0\right)^{\mathsf{T}}$ and $Y = \left(Y_1, \ldots, Y_p, Y_0\right)^{\mathsf{T}}$ with $X_1, \ldots, X_p, X_0 \in \mathbb{R}^n$ and $Y_1, \ldots, Y_p, Y_0 \in \mathbb{R}^n$, we have

$$\left| F\left(X\right) - F\left(Y\right) \right| \leq T \max_{i=0}^{p} \left(\left| X_i - Y_i \right| \right)$$

with

$$T = \frac{1}{2} \left(\sum_{i=1}^{p} E_i \left| I - M_i^{-1} A \right| + \left| I - R_0^{\mathsf{T}} A_0^{-1} R_0 A \right| \right),$$

and where the operator $\max(x, y)$ gives the vector which each entry is the maximum between the corresponding entries in x and y.

If A is an M-matrix, then $I - M_i^{-1}A \geq 0$ for any $i \in \{1, \ldots, p\}$ (see, e.g., Proof of Theorem 4.4 in [25]), therefore, with $I - R_0^\mathsf{T} A_0^{-1} R_0 A \geq 0$ and the multisplitting form (17), we have

$$T = I - \frac{1}{2}\left(M + R_0^\mathsf{T} A_0^{-1} R_0\right) A.$$

If, additionally, A_0 is an M-matrix, then we have $R_0^\mathsf{T} A_0^{-1} R_0 \geq 0$ and hence, by Theorem 4.3 of [25], $\rho(T) < 1$. It follows that, by Lemma 3.1 of [26], the contraction condition for convergence in Theorem 2 of [3] is satisfied for the global iteration mapping, which concludes the proof. □

For some possible constructions of R_0 so as to satisfy the conditions in Theorem 1, see, e.g., [12] and references therein. The asynchronous iterations (12), (14), (15) can be readily implemented following any of the approaches [27–30]. In particular, the implementation of the special formulation (14) can actually be derived from Algorithm 3 of [22] with no effort. The only, but key, difference here is that the solution vector which is corrected is the one from which the RHS of the coarse problem was assembled.

4 Results

In view of a comparison with [12, 22], we shall illustrate the validity of our approach by considering the Poisson's equation,

$$-\Delta u = g, \tag{18}$$

where the unknown solution, u, was defined on a three-dimensional domain $[0, 1] \times [0, 1] \times [0, 0.2]$, and the source term, $g = 918$, was an arbitrary constant. A solution $u = 0$ was prescribed on the boundaries $\partial[0, 1] \times [0, 1] \times [0, 0.2]$ and $[0, 1] \times \partial[0, 1] \times [0, 0.2]$. Each interval of the domain was partitioned such that a discrete overlap of two mesh steps was obtained. The problem (18) was approximated on each subdomain using the P1 finite elements method, which generated the matrices $R_i A$ (the nonzero columns) and vectors $R_i b$, $i \in \{1, \ldots, p\}$. Note that A_i is included in $R_i A$. The global number of unknowns was $n = 174 \times 174 \times 36 = 1,089,936$. The aggregation mapping R_0 was of the form

$$R_0 = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & \cdots & 1 \end{bmatrix},$$

which is a two subdomains example. Cholesky factorization was applied to A_i, $i \in \{1, \ldots, p\}$, and A_0.

The compute platform was a homogeneous cluster of 40 nodes interconnected by a 100 Gb/s Omni-Path Architecture (OPA) network device. Each compute node was provided with 175 GB of memory and two 2.1 GHz processors with 20 cores (40 cores per node). A standard Message Passage Interface (MPI) library was used for the communications. Each core was mapped to one MPI process, and each MPI process was mapped to one subdomain, and vice versa. The coarse problem was solved on the MPI process of rank 0.

In the following, we show the performance scaling of the two-level Schwarz-type solver (6) (referred to as "Sync"), its asynchronous variants "GBCRR" [11] (initials of the authors) and "GM" [22], and the hybrid synchronous-asynchronous variant (12), (14), (15) ("Sync/Async"). The entries of the weighting matrices W_i in (3) were set to 1 or 0 only, so as to have M resulting in the restricted additive Schwarz preconditioner [17]. Overall execution times are compared for a stopping criterion $\|b - Ax\| < \varepsilon\|b\|$ with $\varepsilon = 10^{-6}$. In the asynchronous cases, convergence is detected using non-blocking exact global residual computation, as described in Listing 5 of [29].

In [11, 12], two-level iteration was performed only when a new coarse solution was computed. In the meantime, one-level iterations had to be considered. In [22], several two-level iterations could be performed as long as the gap between the fine and the coarse solutions was within a certain delay interval. We shall therefore consider a parameter, say δ, which corresponds to the maximum number of two-level iterations allowed using the same coarse solution. Table 1 and Figure 1 show the performance of the solvers for an increasing number of processor cores, $p \in \{5 \times 5 \times 1,\ 5 \times 5 \times 2,\ 10 \times 5 \times 2,\ 10 \times 10 \times 2,\ 10 \times 10 \times 4,\ 20 \times 10 \times 4\}$.

p	Sync Eq. (6)	Async, $\delta = 1$ GBCRR [11]	Async, $\delta = 5$ GM [22]	Sync/Async, $\delta = \infty$ Eq. (12), (14), (15)
25	33	44	34	85
50	20	29	19	47
100	11	15	8	23
200	8	8	3	12
400	11	8	3	12
800	27	7	3	17

p	Async, $\delta = 1$ GM [22]	Async, $\delta = \infty$ GM [22]	Sync/Async, $\delta = 1$ Eq. (12), (14), (15)	Sync/Async, $\delta = 5$ Eq. (12), (14), (15)
25	47	34	65	85
50	31	20	41	48
100	17	8	22	23
200	10	3	13	12
400	8	49	11	12
800	7	diverged	11	11

Table 1: Solvers elapsed time in seconds. The parameter δ is the maximum number of two-level iterations per coarse solution.

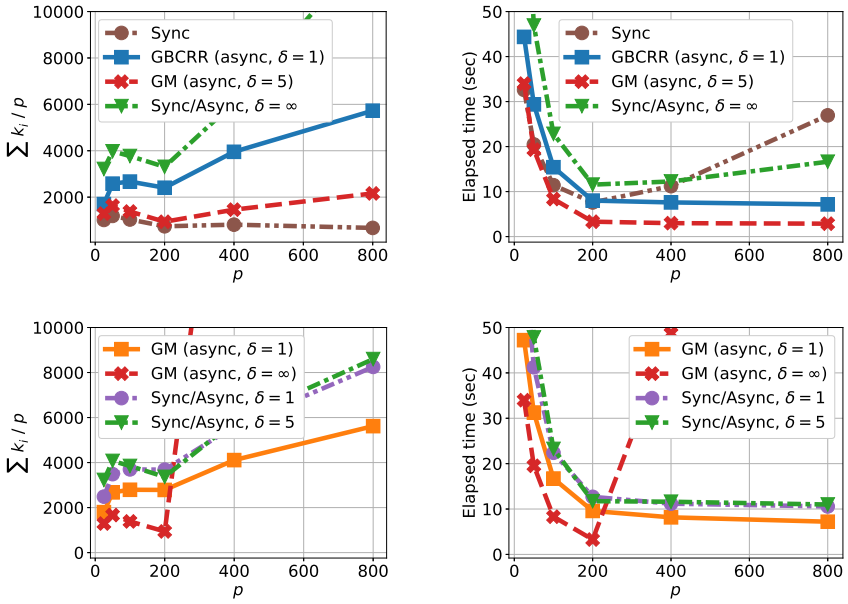

Figure 1: Solvers elapsed time in seconds and average number of iterations (k_i with $i \in \{1, \ldots, p\}$, is the number of iterations on the i-th subdomain), for numbers of processor cores $p \in \{25, 50, 100, 200, 400, 800\}$. The parameter δ is the maximum number of two-level iterations per coarse solution.

5 Conclusions

The development of competitive asynchronous iterative solvers is still at an early stage, however, promising results were achieved through the development of two-level asynchronous domain decomposition methods within both additive Schwarz and primal substructuring frameworks. In this paper, we focused on a new formulation that allows for convergence conditions independent of communications delays. But more importantly, by combining synchronous and asynchronous iterations, this formulation revealed a new exciting perspective. It hopefully suggests the possibility to accelerate asynchronous iterations using highly effective relaxation coefficients computed from synchronized data. However, a difficulty already appears here from the numerical experiments since, while guaranteeing convergence independently of communications delays, the performance of the hybrid synchronous-asynchronous two-level solver seems to have been hindered by the negative impact of those delays on its synchronous part. Efforts should probably now be directed to more efficient combinations of synchronous and asynchronous iterations.

References

[1] J.L. Rosenfeld, "A Case Study in Programming for Parallel-Processors", *Commun. ACM*, 12(12): 645–655, 1969.

[2] D. Chazan, W. Miranker, "Chaotic Relaxation", *Linear Algebra Appl.*, 2(2): 199–222, 1969.

[3] G.M. Baudet, "Asynchronous Iterative Methods for Multiprocessors", *J. ACM*, 25(2): 226–244, 1978.

[4] D.P. Bertsekas, "Distributed asynchronous computation of fixed points", *Math. Program.*, 27(1): 107–120, 1983.

[5] A. Frommer, D.B. Szyld, "On asynchronous iterations", *J. Comput. Appl. Math.*, 123(1–2): 201–216, 2000.

[6] P. Spiteri, "Parallel asynchronous algorithms: A survey", *Adv. Eng. Softw.*, 149: 102896, 2020.

[7] U. Küttler, W.A. Wall, "Fixed-point fluid-structure interaction solvers with dynamic relaxation", *Comput. Mech.*, 43(1): 61–72, 2008.

[8] K. Stüben, "Algebraic multigrid (AMG): experiences and comparisons", *Appl. Math. Comput.*, 13(3): 419–451, 1983.

[9] D. Evans, W. Deren, "An asynchronous parallel algorithm for solving a class of nonlinear simultaneous equations", *Parallel Comput.*, 17(2): 165–180, 1991.

[10] A. Frommer, H. Schwandt, D.B. Szyld, "Asynchronous Weighted Additive Schwarz Methods", *Electron. Trans. Numer. Anal.*, 5: 48–61, 1997.

[11] C. Glusa, E.G. Boman, E. Chow, S. Rajamanickam, P. Ramanan, "Asynchronous One-Level and Two-Level Domain Decomposition Solvers", in R. Haynes, S. MacLachlan, X.C. Cai, L. Halpern, H.H. Kim, A. Klawonn, O. Widlund (Editors), *Domain Decomposition Methods in Science and Engineering XXV*, pages 134–142. Springer International Publishing, Cham, Switzerland, 2020.

[12] C. Glusa, E.G. Boman, E. Chow, S. Rajamanickam, D.B. Szyld, "Scalable Asynchronous Domain Decomposition Solvers", *SIAM J. Sci. Comput.*, 42(6): C384–C409, 2020.

[13] G. Gbikpi-Benissan, F. Magoulès, "Asynchronous Multiplicative Coarse-Space Correction", *SIAM J. Sci. Comput.*, 44(3): C237–C259, 2022.

[14] G. Gbikpi-Benissan, F. Magoulès, "Asynchronous multisplitting-based primal Schur method", *J. Comput. Appl. Math.*, 425: 115060, 2023.

[15] D.J. Evans, S. Jianping, K. Lishan, "The convergence factor of the parallel Schwarz overrelaxation method for linear systems", *Parallel Comput.*, 6(3): 313–324, 1988.

[16] P.L. Lions, "On the Schwarz Alternating Method. I", in R. Glowinski, G.H. Golub, G.A. Meurant, J. Périaux (Editors), *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42. SIAM, Philadelphia, PA, USA, 1988.

[17] X.C. Cai, M. Sarkis, "A Restricted Additive Schwarz Preconditioner for General Sparse Linear Systems", *SIAM J. Sci. Comput.*, 21(2): 792–797, 1999.

[18] J.S. Przemieniecki, "Matrix structural analysis of substructures", *AIAA Journal*, 1(1): 138–147, 1963.

[19] M.J. Gander, "Schwarz methods over the course of time", *Electron. Trans. Numer. Anal.*, 31: 228–255, 2008.

[20] A. Toselli, O. Widlund, *Domain Decomposition Methods - Algorithms and Theory*, Volume 34 of *Springer Series in Computational Mathematics*, Springer-Verlag Berlin Heidelberg, 2005, page 450.

[21] V. Dolean, P. Jolivet, F. Nataf, *An Introduction to Domain Decomposition Methods – Algorithms, Theory, and Parallel Implementation*, SIAM, Philadelphia, PA, USA, 2015, pages IX, 233.

[22] G. Gbikpi-Benissan, F. Magoulès, "Accurate coarse residual for two-level asynchronous domain decomposition methods", in P. Iványi, F. Magoulès, B.H.V. Topping (Editors), *Proceedings of the Seventh International Conference on Parallel, Distributed, GPU and Cloud Computing for Engineering*, Volume CCC 4. Civil-Comp Press, Edinburgh, UK, 2023, Paper 1.1, doi:10.4203/ccc.4.1.1.

[23] G. Karypis, V. Kumar, "Multilevel k-way Partitioning Scheme for Irregular Graphs", *J. Parallel Distrib. Comput.*, 48(1): 96–129, 1998.

[24] A. Frommer, D.B. Szyld, "Weighted max norms, splittings, and overlapping additive Schwarz iterations", *Numer. Math.*, 83(2): 259–278, 1999.

[25] A. Frommer, D.B. Szyld, "An Algebraic Convergence Theory for Restricted Additive Schwarz Methods Using Weighted Max Norms", *SIAM J. Numer. Anal.*, 39(2): 463–479, 2001.

[26] G. Gbikpi-Benissan, F. Magoulès, "Resilient asynchronous primal Schur method", *Appl. Math.*, 67: 679–704, 2022.

[27] M. Chau, D. El Baz, R. Guivarch, P. Spiteri, "MPI implementation of parallel subdomain methods for linear and nonlinear convection-diffusion problems", *J. Parallel Distrib. Comput.*, 67(5): 581–591, 2007.

[28] J. Wolfson-Pou, E. Chow, "Reducing Communication in Distributed Asynchronous Iterative Methods", *Procedia Computer Science*, 80: 1906–1916, 2016.

[29] F. Magoulès, G. Gbikpi-Benissan, "JACK2: An MPI-based communication library with non-blocking synchronization for asynchronous iterations", *Adv. Eng. Softw.*, 119: 116–133, 2018.

[30] I. Yamazaki, E. Chow, A. Bouteiller, J. Dongarra, "Performance of asynchronous optimized Schwarz with one-sided communication", *Parallel Comput.*, 86: 66–81, 2019.