Asynchronous Domain Decomposition Methods for Nonlinear PDEs

Fayçal Chaouqui, Edmond T. Chow
and Daniel B. Szyld

Report 21-08-16
August 2021. Revised July 2022

Department of Mathematics
Temple University
Philadelphia, PA 19122

This report is available in the World Wide Web at
http://www.math.temple.edu/~szyld

ASYNCHRONOUS DOMAIN DECOMPOSITION METHODS FOR NONLINEAR PDES*

FAYÇAL CHAOUQUI†, EDMOND CHOW‡, AND DANIEL B. SZYLD§

Abstract. One- and two-level parallel asynchronous methods for the numerical solution of nonlinear systems of equations, especially those arising from (nonlinear) partial differential equations, are studied. The proposed methods are based on domain decomposition techniques. Local convergence theorems are presented in several cases, with appropriate hypotheses. Computational results on a shared memory multiprocessor machine for various problems exhibiting nonlinearities are reported, illustrating the potential of these asynchronous methods, especially for heterogeneous clusters.

Key words. Asynchronous iterations. Nonlinear problems. Domain decomposition. Partial Differential Equations. Two-level methods.

AMS subject classifications. 65N22, 65F08, 65F10, 65F50

1. Introduction.

Asynchronous iterative methods are currently undergoing a resurgence in popularity due to the dramatic increase of parallelism in modern computers. The good performance of asynchronous methods is due to not needing to synchronize the computational tasks, hence minimizing idle time, i.e., time for which some processors are inactive but can be used. The effect of data exchange in the asynchronous method is then less pronounced; see, e.g., [2, 4, 6, 22]. In these methods, the iterations (or updates) are carried out in parallel by processors in an arbitrary order and without any synchronization. These asynchronous iterative methods are especially attractive when the network of processors is heterogeneous, or the communication costs are large, or when the differences processors have different loads, e.g., when each processor solves local problems with different physical properties or with different type of nonlinearities.

In this paper, we study the parallel solution of nonlinear systems of equations. These often arise from the finite element or finite different discretization of nonlinear differential equations such as nonlinear convection-diffusion problems. We present new one- and two-level algorithms based on domain decomposition methods. The domain of the differential equation is subdivided into several possibly overlapping subdomains, and the computational tasks are assigned to different processors whereby the local components of the iterate vector can be updated without any order nor synchronization.

Our point of departure is the restricted additive Schwarz (RAS) iterative method for linear systems [12], which we review in Section 2. We then introduce a restricted additive Schwarz method for nonlinear problems, and analyze its local convergence properties. We note that RAS has been used as preconditioners for the linear systems arising in Newton's methods for nonlinear problems, e.g., in [10, 11, 17, 30]. In these methods, the linear systems at each step are solved using a Krylov subspace method and preconditioned with RAS. Since we have in mind parallel computers with high communication costs, the use of Krylov methods may not be suitable, since each iteration usually entails orthogonalizations with the concomitant inner products,

*This version dated July 23, 2022. This research is supported in part by the U.S. Department of Energy under grant DE-SC0016578.

†Department of Mathematics,Temple University, Philadelphia, PA, USA, chaouqui@temple.edu

‡College of Computing, Georgia Institute of Technology, Atlanta, GA, USA, echow@cc.gatech.edu

§Department of Mathematics, Temple University, Philadelphia, PA, USA, szyld@temple.edu

which entail communication among all processors. This is why we concentrate on RAS as a solver, not as a preconditioner. We note that earlier work has been done on using RAS as a preconditioner; cf. [17, 13]. See also the recent work [29] for two-level RAS preconditioners using additive/multiplicative coupling of coarse spaces, and [15] for substructured coarse correction, and [24] for asynchronous multiplicative coarse correction.

Our contribution includes the introduction of the asynchronous version of the nonlinear restricted additive Schwarz method in Section 4, where we also state convergence results, and discuss its implementation. We further introduce a coarse grid correction in Section 5, and discuss convergence results of the two-level method. Throughout the paper we illustrate the performance of the methods with numerical experiments.

2. Domain Decomposition Methods. Our goal is to study the parallel solution of discretized general nonlinear elliptic equations in $\mathbb{R}^d$ ($d \geq 2$) of the form

$$L(u) \colon = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} a_i(x, u(x), \nabla u(x)) + a_0(x, u(x), \nabla u(x)) = f(x), \quad \text{in } \Omega, \tag{2.1}$$

with homogeneous Dirichlet boundary conditions, i.e., $u = 0$ on $\partial\Omega$. Here Ω is a bounded region in $\mathbb{R}^d$ with Lipschitz continuous boundary. Let $p = (p_0, p_1, \ldots, p_d) \in \mathbb{R}^{d+1}$ be such that one can define the functions a_i as $a_i(x, u(x), \nabla u(x)) = a_i(x, p)$, for $i = 0, \ldots, d$. Existence and uniqueness of a solution to problem (2.1) can be stated under certain hypotheses on the coefficients $a_i(x, p)$ as follows. Let $c, C > 0$ be such that for $i = 0, \ldots, d$, $a_i(x, p)$ satisfy

(i) $a_i \in C^1(\Omega \times \mathbb{R}^{d+1})$

(ii) $\max\left\{|a_i|, \left|\frac{\partial a_i}{\partial x_j}\right|, \left|\frac{\partial a_i}{\partial p_k}\right|\right\} \leq C$, for $i, k = 0, \ldots, d$, and $j = 1, \ldots, d$

(iii) $\sum_{i,j=0}^{d} \frac{\partial a_i(x,p)}{\partial p_j} \xi_i \xi_j \geq c \sum_{i=0}^{d} \xi_i^2$ for $(\xi_0, \ldots, \xi_d) \in \mathbb{R}^{d+1}$.

Then, the problem (2.1) is well posed in $H^1(\Omega)$, where $H^1(\Omega)$ is the space of functions $v \in L^2(\Omega)$ such that $\nabla v \in L^2(\Omega)$, and the space is equipped with the norm $\|v\|^2_{H^1(\Omega)} := \|v\|^2_{L^2(\Omega)} + \|\nabla v\|^2_{L^2(\Omega)}$; see, e.g., [31].

The problem (2.1) after discretization with, e.g., finite differences or finite elements, yields a large nonlinear system of equations of the form

$$F(u) = B(u) - f = 0, \tag{2.2}$$

where $B, F : V \mapsto V$, $V := \mathbb{R}^n$, and n is total number of unknowns. Here, we abuse the notation and keep the same symbol for the continuous and discrete variables. Throughout this article, we denote by $u^* \in \mathbb{R}^n$ the vector that represents the solution of (2.2).

2.1. Restricted Additive Schwarz (RAS) for linear systems. We introduce in the next section the restricted Additive Schwarz (RAS) method for nonlinear equations. We begin here by reviewing RAS for the case that $F(u)$ is linear. Let us thus consider

$$F(u) = Au - b, \tag{2.3}$$

where $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$. We assume that F has a simple zero u^*, i.e., that A is nonsingular, or in other words, A is of full rank, i.e., $\operatorname{rank} A = n$. The restricted

additive Schwarz introduced in [12] provides a means of constructing parallel iterative solvers based on domain decomposition techniques. The general philosophy of RAS is that the local problems (i.e., in each subdomain) are solved with overlap, but the variables communicated to the other local problems are those corresponding to subdomains without overlap.

Let $S = \{1, \ldots, n\}$ be the set of indices of all the variables. Let the sets $S_i \subset S$, $i = 1, \ldots, P$, form a non-overlapping partitioning, i.e.,

$$\bigcup_{i=1}^{P} S_i = S, \quad S_i \cap S_j = \emptyset, \ i = 1, \ldots, P.$$

These correspond to the variables of the subdomains without overlap. For the overlapping subdomains, for any $\delta > 0$, we define $S_i \subset S_{i,\delta} \subset S$, $i = 1, \ldots, P$, thus defining a covering of S. We set $V_{i,\delta} := \mathbb{R}^{n_{i,\delta}}$ where $n_{i,\delta} = |S_{i,\delta}|$, the cardinality of the set of variables corresponding to the ith subdomain with overlap δ. We define also the restriction mappings $R_{i,\delta} \colon V \mapsto V_{i,\delta}$, $i = 1, \ldots, P$. These restriction operators satisfy the identities

$$\begin{aligned} R_{i,\delta} R_{i,\delta}^\top &= I_{V_{i,\delta}} \\ R_{i,\delta}^\top R_{i,\delta|R_{i,\delta}^\top(V_{i,\delta})} &= I_{|R_{i,\delta}^\top(V_{i,\delta})}. \end{aligned}$$

Moreover, we have

$$\sum_{i=1}^{P} R_{i,0}^\top R_{i,\delta} = I,$$

which is known as the partition of unity condition.

We are ready to define the RAS iterative method. Given an initial vector u^0, the iteration is given by

$$u^{k+1} = u^k + \sum_{i=1}^{P} R_{i,0}^\top A_{i,\delta}^{-1} R_{i,\delta}(b - Au^k), \tag{2.4}$$

where $A_{i,\delta} := R_{i,\delta} A R_{i,\delta}^\top \colon V_{i,\delta} \mapsto V_{i,\delta}$ is the coefficient matrix for the local problem. Observe in (2.4) how the local problem is solved with the overlap, but only the variables without the overlap contribute to the next iterate. The iteration operator corresponding to (2.4) is thus given by

$$T_{A,\delta} = I - \sum_{i=1}^{P} R_{i,0}^\top A_{i,\delta}^{-1} R_{i,\delta} A. \tag{2.5}$$

We present a convergence and comparison result for (linear) RAS from [23]. We need some notation first. Let $\geq$ ($>$) in $\mathbb{R}^n$ and $\mathbb{R}^{n\times n}$ denote the natural elementwise partial ordering, i.e., for $x, y \in \mathbb{R}^n$, $x \geq y$ ($x > y$) if $x_j \geq y_j$ ($x_j > y_j$) for $j = 1, \ldots, n$. Let $w \in \mathbb{R}^n$, $w > 0$, and let $\|\cdot\|_w$ denote the weighted max-norm operator corresponding to the vector norm $\|x\|_w = \max_{j=1,\ldots,n} |x_j/w_j|$. A nonsingular matrix A is called an M-matrix if its off-diagonal elements are nonpositive and the inverse is nonnegative, i.e., $A^{-1} \geq 0$.

THEOREM 2.1 ([23]). *Let A be a nonsingular M-matrix. Let $w > 0$ such that $Aw > 0$. Then, if $\delta \geq \delta'$,*

$$\|T_{A,\delta}\|_w \leq \|T_{A,\delta'}\|_w < 1.$$

Theorem 2.1 shows that the (linear) RAS iterative method case converges linearly, and that the larger the overlap, the faster the asymptotic convergence. Recall that the classical additive Schwarz method may fail to converge as an iterative method. The main difference is that here the variables of the overlap are not included in the new iterate (and thus there is no "double counting"); see [19, 23].

For completeness, we remark that when RAS is used as a preconditioner for Krylov subspace methods, although this is not the focus of this paper, it follows from (2.5) that the action of the RAS preconditioner M^{-1} on the global matrix A is given by

$$M^{-1}A = \sum_{i=1}^{P} R_{i,0}^{\top} A_{i,\delta}^{-1} R_{i,\delta} A.$$

As mentioned in the introduction, we concentrate in this paper on asynchronous parallel solvers, and therefore we do not consider Krylov subspace methods (or preconditioning), since the orthogonalization processes lead to synchronization points.

2.2. Nonlinear Restricted Additive Schwarz (NLRAS). We return to the case of nonlinear F and now describe the nonlinear iterative RAS (NLRAS) algorithm. While this algorithm is new, we base our presentation on [18], where general Schwarz methods for nonlinear problems are discussed. Define the restriction operators

$$\begin{aligned} F_{i,\delta} : V &\mapsto V_{i,\delta} \\ v &\mapsto R_{i,\delta} F(v), \end{aligned}$$

for $i = 1, \ldots, P$. For any given $u = u^k \in \mathbb{R}^d$, define $v_{i,\delta}$ as the solution of the local nonlinear problem

$$F_{i,\delta}(u + R_{i,\delta}^{\top} v_{i,\delta}) = 0, \quad \text{for } i = 1, \ldots, P. \tag{2.6}$$

We can then define the NLRAS fixed point iteration as

$$u^{k+1} = u^k + \sum_{i=1}^{P} R_{i,0}^{\top} v_{i,\delta}^k. \tag{2.7}$$

Note here again the philosophy of RAS: we solve the local (nonlinear) problem (2.6) with the overlap, but only consider the non-overlapping variables as a contribution to the next iterate.

We provide now a result showing that (2.7) converges locally and has an asymptotic convergence rate similar to that of the linear case. To this end, let T define the nonlinear mapping

$$T(u) = u + \sum_{i=1}^{P} R_{i,0}^{\top} v_{i,\delta}, \tag{2.8}$$

where $v_{i,\delta}$ is the solution of the local nonlinear problem (2.6). Let u^* be the solution of (2.2). We shall prove that T is a contraction. Hence it admits a fixed point and the iterative process (2.7) convergences to u^*; cf. [16].

THEOREM 2.2. *Suppose that $F'(u^*)$ is a nonsingular M-matrix. Let $w > 0$ be such that $F'(u^*)w > 0$. Then, there exist $0 < \zeta < 1$, and $\mathcal{U}$ a neighborhood of u^* so that*

$$\|T(u) - T(u')\|_w \leq \zeta \|u - u'\|_w \quad \textit{for all } u, u' \in \mathcal{U},$$

i.e., T has a fixed point u^*, and it is the solution of (2.2).

Proof. By definition, we have

$$\begin{aligned} T(u) - T(u') &= \left(u + \sum_{i=1}^{P} R_{i,0}^{\top} v_{i,\delta} \right) - \left(u' + \sum_{i=1}^{P} R_{i,0}^{\top} v'_{i,\delta} \right) \\ &= u - u' + \sum_{i=1}^{P} R_{i,0}^{\top} (v_{i,\delta} - v'_{i,\delta}). \end{aligned} \tag{2.9}$$

Let us define the multidimensional differential quotient of F by

$$DF(v, v') = \int_0^1 F'(v + t(v' - v))\, \mathrm{d}t, \quad v, v' \in \mathbb{R}^n.$$

Let us denote the matrix $J_{i,\delta} := DF(u + R_{i,\delta}^{\top} v_{i,\delta}, u' + R_{i,\delta}^{\top} v'_{i,\delta})$. We have that $v_{i,\delta}$ and $v'_{i,\delta}$ satisfy

$$v_{i,\delta} - v'_{i,\delta} = -(R_{i,\delta} J_{i,\delta} R_{i,\delta}^{\top})^{-1} R_{i,\delta} J_{i,\delta} (u - u'). \tag{2.10}$$

Indeed, using (2.6), we have that $v_{i,\delta}$ and $v'_{i,\delta}$ satisfy

$$\begin{aligned} 0 &= F_{i,\delta}(u + R_{i,\delta}^{\top} v_{i,\delta}) - F_{i,\delta}(u' + R_{i,\delta}^{\top} v'_{i,\delta}) \\ 0 &= R_{i,\delta}(F(u + R_{i,\delta}^{\top} v_{i,\delta}) - F(u' + R_{i,\delta}^{\top} v'_{i,\delta})) \\ 0 &= R_{i,\delta} DF(u + R_{i,\delta}^{\top} v_{i,\delta}, u' + R_{i,\delta}^{\top} v'_{i,\delta})(u - u' + R_{i,\delta}^{\top}(v_{i,\delta} - v'_{i,\delta})), \\ 0 &= R_{i,\delta} J_{i,\delta}(u - u' + R_{i,\delta}^{\top}(v_{i,\delta} - v'_{i,\delta})), \end{aligned}$$

from which we obtain,

$$R_{i,\delta} J_{i,\delta}(u - u') = -R_{i,\delta} J_{i,\delta} R_{i,\delta}^{\top}(v_{i,\delta} - v'_{i,\delta}).$$

Hence, substituting (2.10) in (2.9), we obtain

$$\begin{aligned} T(u) - T(u') &= u - u' - \sum_{i=1}^{P} R_{i,0}^{\top} (R_{i,\delta} J_{i,\delta} R_{i,\delta}^{\top})^{-1} R_{i,\delta} J_{i,\delta} (u - u') \\ &= \left(I - \sum_{i=1}^{P} R_{i,0}^{\top} (R_{i,\delta} J_{i,\delta} R_{i,\delta}^{\top})^{-1} R_{i,\delta} J_{i,\delta} \right) (u - u'). \end{aligned} \tag{2.11}$$

To complete the proof, we remark that by the definition of DF,

$$(R_{i,\delta} DF(u, u') R_{i,\delta}^{\top})^{-1} R_{i,\delta} DF(u, u') \to (R_{i,\delta} F'(u^*) R_{i,\delta}^{\top})^{-1} R_{i,\delta} F'(u^*) \text{ as } u, u' \to u^*. \tag{2.12}$$

Moreover, formula (2.10) shows that $v_{i,\delta}, v'_{i,\delta} \to 0$, when $u, u' \to u^*$. From this it follows that

$$I - \sum_{i=1}^{P} R_{i,0}^{\top} (R_{i,\delta} J_{i,\delta} R_{i,\delta}^{\top})^{-1} R_{i,\delta} J_{i,\delta} \to T_{F'(u^*),\delta} \text{ as } u, u' \to u^*,$$

where we used the notation in (2.5). This shows that for $\|T_{F'(u^*),\delta}\|_w < \zeta < 1$, there exists a neighborhood $\mathcal{U}$ of u^* such that

$$\left\| I - \sum_{i=1}^{P} R_{i,0}^{\top}(R_{i,\delta}J_{i,\delta}R_{i,\delta}^{\top})^{-1}R_{i,\delta}J_{i,\delta} \right\|_w \leq \zeta < 1,$$

It suffices then to take the norm $\|\cdot\|_w$ on both sides of (2.11) to obtain the stated result. □

Theorem 2.2 shows that the iterative procedure $u^{k+1} = T(u^k)$ for $u^0 \in \mathcal{U}$ converges to $u = T(u)$ for some u. This implies that we converge to the point $u = u^*$, the unique solution of (2.2). Moreover, the same result shows that the mapping T in (2.8) is differentiable on $u = u^*$, with $T'(u^*) = T_{F'(u^*),\delta}$. This also demonstrates that NLRAS with an overlap $\delta' \geq \delta$ has in general a faster local convergence rate since $\|T_{F'(u^*),\delta'}\|_w \leq \|T_{F'(u^*),\delta}\|_w$ (see Theorem 2.1) .

We remark that the hypothesis that $F'(u^*)$ is a nonsingular M-matrix may or may not be satisfied for many practical problems. While we can only prove convergence in this case, numerical experiments indicate that the conclusion of the theorem is valid for a wider class of problems; see, e.g., the numerical examples in Section 4.

Note that as domain decomposition methods for the linear case can be regarded as a way to construct preconditioners for Krylov methods, nonlinear domain decomposition methods such as NLRAS can be a way to construct nonlinear preconditioners for the Newton method applied to (2.2), and for which the following equivalent problem

$$\mathcal{F}(u) := \sum_{i=1}^{P} v_{i,\delta} = 0$$

is solved. However, we do not pursue this idea here, and we refer the interested reader to, e.g., [10, 11, 17, 30].

3. General Asynchronous iterations. Asynchronous iterations refer to a class of parallel iterative procedures in which each processor executes its computations at the next iteration without waiting for the others to finish theirs. That is, there is no synchronization. The concept of iteration thus loses its meaning, since different processors update elements of the global approximation at different times (these are called timestamps below). Thus some processors use information which may have been updated in some processors more times than in others.

Mathematical models of asynchronous iterations were developed in order to study their convergence; see, e.g., [2, 4, 6, 22, 34, 36]. Sometimes these models are referred to as mathematical descriptions of computational models of asynchronous iterations. We review one such model from [4], which is now classic. Let $U = U_1 \times \ldots \times U_P$, $T : U \mapsto U$, and $T_i(u) = (Tu)_i$. The goal is to solve (2.2) by way of the equivalent fixed point problem $u = T(u)$ in parallel, so that processor i runs iterations on the ith local problem, i.e., $u_i^{new} = T_i(u)$. Let us call a *timestamp* the instant of time at which at least one processor finishes its computation and updates its associated local variables. Define by t_k the sequence of all these timestamps, and by $s_j^i(k)$ the sequence of integers that represent the timestamp index of the local variables coming from processor j, and that are available to processor i at time instant when they start solving the local problem which is completed at time t_k. Let us also denote by I^k a subset of $\{1, \ldots, P\}$ that defines the list of subdomains that are being updated at timestamp t_k.

To solve $u = T(u)$, we generate asynchronously a sequence of vectors $\{u_i^k\}_{k\geq 0}$ for $i = 1, \dots P$, satisfying

$$u_i^{k+1} = \begin{cases} T_i(\dots, u_j^{s_j^i(k)}, \dots) & \text{for } i \in I^k \\ u_i^k & \text{for } i \notin I^k, \end{cases} \tag{3.1}$$

that is, either the ith portion of the solution is updated with the results of the computations in the ith processor at the timestamp t_k, or it is not updated. We also make the following three assumptions.

(i) $\forall i, j \in \{1, \dots, P\}, \quad \forall k \in \mathbb{N}, s_j^i(k) \leq k$,

(ii) $\forall i, j \in \{1, \dots, P\}, \quad \lim_{k\to\infty} s_j^i(k) = +\infty$,

(iii) $\forall i \in \{1, \dots, P\}, \quad \big|\{k \geq 0 \colon i \in I^k\}\big| = +\infty$.

The first assumption makes sure that no future updates are being used at the beginning of the computation. The second ensures that eventually each processor is receiving new information. The last one indicates that no processor stops being updated. Convergence results for (3.1) were discussed and analyzed in several papers; see, e.g., [2, 22, 34] and references therein.

We list below three convergence results for general asynchronous iterations, which we use in our analysis of the methods discussed in this paper.

THEOREM 3.1. ([22, Theorem 4.1] following [4]) *Let T be an $n \times n$ matrix, i.e., representing a linear map. Assume that the conditions* (i)–(iii) *hold. If the spectral radius $\rho(|T|) < 1$, where $|T|$ has components $|t_{ij}|$, then the asynchronous iteration* (3.1) *converges to the solution of the fixed point problem $u = Tu$.*

For example, when T is the RAS operator (2.5) and A is a nonsingular M-matrix, we know from Theorem 2.1 that there exists a vector $w > 0$ such that $\|T_{A,\delta}\|_w < 1$. This implies that $\rho(|T_{A,\delta}|) < 1$, and we can use Theorem 3.1 to show that asynchronous RAS converges.

THEOREM 3.2. ([22, Theorem 4.4] following [20]) *Assume that u^*, the solution of* (2.2)*, i.e., the fixed point of T, lies in the interior of U and that T is Fréchet differentiable at u^*. Assume that the conditions* (i)–(iii) *hold and that $\rho(|T'(u^*)|) < 1$, then there exists a neighborhood $\mathcal{U}$ of u^*, such that the asynchronous iteration* (3.1) *converges to u^* for any $u^0 \in \mathcal{U}$.*

THEOREM 3.3. ([22, Theorem 3.3] following [20]) *Let T^k be a sequence of (nonlinear) operators, $T^k : U \to U$, such that there exists a common fixed point $u^* \in U$ for which $T^k(u^*) = u^*$ for all k. Moreover, assume that there exist $0 \leq \zeta < 1$, and $w \in \mathbb{R}^n$, $w > 0$ such that*

$$\|T^k(u) - u^*\|_w \leq \zeta \|u - u^*\|_w,$$

for all $k \geq 0$. Assume that the conditions (i)–(iii) *hold. Then, the asynchronous iteration*

$$u_i^{k+1} = \begin{cases} T_i^k(\dots, u_j^{s_j^i(k)}, \dots) & \text{for } i \in I^k \\ u_i^k & \text{for } i \notin I^k, \end{cases} \tag{3.2}$$

converges to u^*, the unique fixed point of all the operators T^k.

4. Asynchronous nonlinear RAS. Asynchronous parallel methods for nonlinear equations have been studied before; see, e.g. [3, 7, 14, 21, 25, 32, 35, 37]. The experimental results obtained in these papers showed that the asynchronous methods

perform better in terms of execution times than their synchronous counterparts; see e.g., [14, Tables C5–C7]. Here we present the asynchronous RAS method for nonlinear problems, provide local convergence results, and compare its performance with that of the synchronous counterpart. We emphasize that our approach is well suited to heterogeneous networks, and in general to parallel computers where synchronization and communication are at a premium.

The asynchronous nonlinear RAS method consists of the standard asynchronous iteration (3.1), where T_i is given by (2.6) and (2.8); see also Algorithm 1 below. We begin with a local convergence theorem.

THEOREM 4.1. *Let u^* be the solution of* (2.2)*. Suppose that F' exists on a neighborhood of u^* and that $F'(u^*)$ is a nonsingular M-matrix. Assume that the conditions* (i)–(iii) *hold. Then, there exists a neighborhood $\mathcal{U}$ of u^* such that asynchronous nonlinear RAS (NLRAS) defined by* (3.1) *with T_i is given by* (2.6) *and* (2.8)*, converges to u^* for any initial vector $u^0 \in \mathcal{U}$.*

Proof. The proof is essentially the same as that in Theorem 2.2. However, here we need the additional hypothesis that T is differentiable on a region containing u^*. We recall that the nonlinear RAS iteration is given by

$$u^{k+1} = T(u^k), \quad T(u) = u + \sum_{i=1}^{P} R_{i,0}^{\top} v_{i,\delta},$$

where $F_{i,\delta}(u + R_{i,\delta}^{\top} v_{i,\delta}) = 0$. To prove that the asynchronous iteration associated with the operator T converges, it suffices to prove that $\|T'(u^*)\| < 1$ for some operator norm. To this end, we compute

$$T'(u) = I + \sum_{i=1}^{P} R_{i,\delta}^{\top} \frac{\partial v_{i,\delta}}{\partial u}.$$

Also, differentiating $F_{i,\delta}$, we obtain

$$F'_{i,\delta}(u + R_{i,\delta}^{\top} v_{i,\delta})(I + R_{i,\delta}^{\top} \frac{\partial v_{i,\delta}}{\partial u}) = 0,$$

hence,

$$\begin{aligned} F'_{i,\delta}(u + R_{i,\delta}^{\top} v_{i,\delta}) R_{i,\delta}^{\top} \frac{\partial v_{i,\delta}}{\partial u} &= -F'_{i,\delta}(u + R_{i,\delta}^{\top} v_{i,\delta}) \\ \frac{\partial v_{i,\delta}}{\partial u} &= -\left(R_{i,\delta} F'(u + R_{i,\delta}^{\top} v_{i,\delta}) R_{i,\delta}\right)^{-1} F'_{i,\delta}(u + R_{i,\delta}^{\top} v_{i,\delta}). \quad (4.1) \end{aligned}$$

Thus,

$$T'(u) = I - \sum_{i=1}^{P} R_{i,\delta}^{\top} \left(R_{i,\delta} F'(u + R_{i,\delta} v_{i,\delta}) R_{i,\delta}\right)^{-1} F'_{i,\delta}(u + R_{i,\delta}^{\top} v_{i,\delta}). \quad (4.2)$$

Evaluating (4.2) at $u = u^*$, we obtain

$$T'(u^*) = I - \sum_{i=1}^{P} R_{i,\delta}^{\top} \left(R_{i,\delta} F'(u^*) R_{i,\delta}\right)^{-1} F'_{i,\delta}(u^*) = T_{F'(u^*),\delta},$$

where the latter is exactly the same as in (2.5) but applied to $F'(u^*)$, hence $\|T'(u^*)\|_w = \|T_{F'(u^*),\delta}\|_w < 1$ for $w > 0$ such that $F'(u^*)w > 0$. Using Theorem 3.2, with the observation that $\rho(|T_{F'(u^*),\delta}|) \leq \|T_{F'(u^*),\delta}\|_w$, we conclude that there exists a neighborhood of $\mathcal{U}$ of u^* such that the asynchronous nonlinear RAS is convergent. □

As was the case for the synchronous method, the convergence proof relies on the hypothesis that $F'(u^*)$ is a nonsingular M-matrix. Many of the problems in practice do not satisfy this hypothesis, but as we illustrate with the numerical experiments below, the conclusion of the theorem still holds in practice.

We describe next the asynchronous stopping criterion we used in our implementation of asynchronous NLRAS on a shared memory machine. For the sake of simplicity, we assume that each computational thread is assigned to a unique subdomain. The goal is to reduce the relative residual norm below a prescribed tolerance $\epsilon > 0$, namely

$$\|F(u^k)\|_2 / \|F(u^0)\|_2 < \epsilon. \tag{4.3}$$

We observe that the L^2-norm of the global residual can be computed as a sum of parts

$$\|F(u)\|_2^2 = \sum_{i=1}^{P} \|F_{i,0}(u)\|_2^2 \cdot$$

Hence, in order to check that the global stopping criterion was achieved, it suffices to check that the local solution on each subdomain satisfies the local convergence criteria $\|F_{i,0}(u)\|_2 / \|F(u^0)\|_2 < \epsilon/\sqrt{P}$. This condition ensures that the contribution of the local residual is such that (4.3) holds. In our implementation, a global boolean variable is declared and is then set to true when all the subdomains satisfy the local convergence criteria. We comment that in order to update the local iteration vectors and residuals, we only need to communicate with the neighboring subdomains. This allows us to check the local stopping criteria and hence avoid global communication between the subdomains/processors.

We describe the implementation of the asynchronous NLRAS in Algorithm 1.

Algorithm 1 Asynchronous nonlinear RAS (ANLRAS)

Require: Initial vector u^0, tolerance ϵ.
Compute $\|F(u^0)\|_2$.
Set global_convergence=**false**, and local_convergence[k]=**false**, $k = 1, \ldots, P$.
In parallel, each processor i:
while not global_convergence **do**
  Compute $v_{i,\delta}$ by solving (2.6) using Newton's method.
  Update $u = u + R_{i,0}^{\top} v_{i,\delta}$
  if $\|F_{i,0}(u)\|_2 / \|F(u^0)\|_2 < \epsilon/\sqrt{P}$ **then**
    Set local_convergence[i]=**true**
  end if
  if i==1 **then**
    if local_convergence[k], $\forall k = 1, \ldots, P$ **then**
      Set global_convergence=**true**
    end if
  end if
end while (for processor i)
Output: u^*.

We mention that once a processor/subdomain achieves local convergence, the local convergence can be lost before the global convergence flag is triggered. This is why we compute the global residual norm again after the iterations stopped and check that it is below the convergence threshold. This was always satisfied in our experiments.

We illustrate the performance of this algorithm on two nonlinear PDEs. Let us consider first the following problem,

$$-\nabla^2 u + g(u) = f \quad \text{on } \Omega, \quad u = 0 \quad \text{on } \partial\Omega, \tag{4.4}$$

where $\Omega = (0,1)^2$ and $g(u) = ue^u$. We choose f such that $\sin(\pi x)\sin(\pi y)$ is the exact solution of (4.4). The discretization of (4.4) using standard 5-points stencil with mesh size h on the nodes (ih, jh) yields to the following nonlinear system of equations

$$(F(u))_{i,j} = \frac{1}{h^2}(4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j+1} - u_{i,j-1}) + u_{i,j}e^{u_{i,j}} - f_{i,j} = 0, \tag{4.5}$$

with $u_{0,j} = u_{N+1,j} = u_{j,0} = u_{j,N+1} = 0$. We present and analyze computational results for parallel asynchronous and synchronous NLRAS for the discrete nonlinear system of equations (4.5). The computational experiments were carried out on a shared memory machine with 88 CPU cores/176 threads and 1536GB of RAM. The parallelization of the solvers was implemented in C++ using the OpenMP multithreading library. The linear algebra data structures and solvers were provided by EIGEN [27]. The local problems in (2.6) were solved using an inner Newton's method where the local convergence criterion is when the norm of the difference between two updates in the local Newton's iterations is smaller than 10^{-10}, and the local Jacobian matrices arising from the inner Newton's iterations were computed analytically. We use a checker-board partitioning for the domain $\Omega = (0,1)^2$ with a total of 10 000 discretization points. We choose the zero vector as an initial vector for both the RAS iteration and the local Newton's iterations.

For each tolerance in $10^{-1}, 10^{-2}, \ldots, 10^{-8}$, we run Algorithm 1 and measure the corresponding computational time (in seconds) required to reach that specific tolerance. We then plot the timing versus the relative residual norm for both methods as shown in Figure 4.1 (left). We observe in this case that the synchronous method is faster in terms of execution time. We repeat the same experiment, but with one processor twice as slow as the others by performing the inner Newton's solve twice for one processor, and report the result in Figure 4.1 (right). It is clear that in this case, the asynchronous method outperforms the synchronous one in terms of execution time. This indicates that even with a small number of processors, if the network is heterogeneous, asynchronous methods may be preferred. The same effect would be observed if there is a large load imbalance, that is, if one or more processors need to perform more work than others.

We report in Table 4.1 the number of local updates for different number of subdomains for both the synchronous and the asynchronous NLRAS. With one processor twice as slow, the asynchronous method is faster than the synchronous method even if, on average, the asynchronous method requires more local updates.

We also compare the number of Newton solves required for each subdomain/processor for both the synchronous and asynchronous cases. We illustrate this in Figure 4.2. We can see from Figure 4.2 that the number of Newton iterations is slightly different

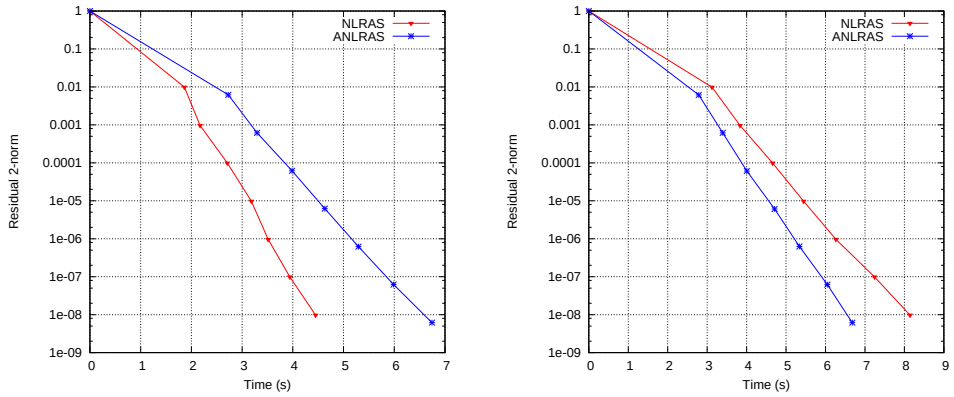

Fig. 4.1: Convergence curves of synchronous and asynchronous NLRAS for problem (4.4) with $P = 16$. Left: Homogeneous network. Right: One processor slower than the others by a factor of two.

#DOF	#subdomains	NLRAS #iter (min,mean,max)	NLRAS time
Synchronous			
2 500	4	160	2.40
5 625	9	323	4.79
10 000	16	554	8.55
Asynchronous			
2 500	4	(149, 250, 287)	2.18
5 625	9	(282, 508, 570)	4.20
10 000	16	(470, 834, 892)	6.89

Table 4.1: Number of iterations/updates and computational time (in seconds) for problem (4.4) where one processor is twice as slow.

for each processor. We also observe that the asynchronous case requires more Newton iterations compared to the synchronous case since it usually requires more local updates.

In order to investigate the influence of the heterogeneity factor on the performance of both the synchronous and asynchronous methods, we perform an experiment where we vary the heterogeneity factor in the interval $[0, 4]$ and compute the timing required to reach a tolerance of 10^{-8}. This can be done by enforcing one processor to wait a given amount of time using the C++ built-in function `sleep_for()` that is proportional to the local Newton solve. The plots in Figure 4.3 illustrate the time required to reach convergence for each method and with different heterogeneity factors. We can observe that there is a heterogeneity threshold for which the asynchronous method starts to outperform the synchronous one. We can also see that the time required for the asynchronous method remains constant while the time required for the syn-

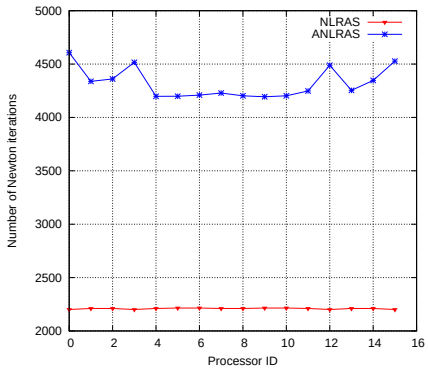

Fig. 4.2: Total number of Newton iterations per processor for problem (4.4) with $P = 16$.

chronous method increases roughly proportionally with the heterogeneity factor.

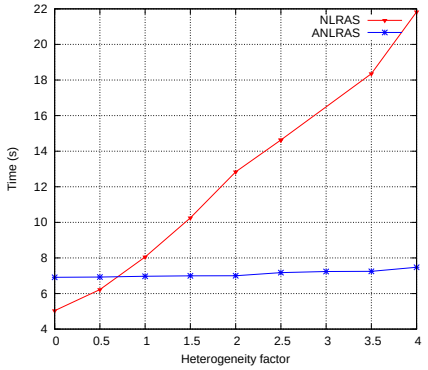

Fig. 4.3: Execution time of synchronous and asynchronous NLRAS with different heterogeneity factors for problem (4.4) with $P = 16$.

For our next set of experiments, we consider a problem where the nonlinearity is on the boundary. This is a temperature control model defined as follows:

$$\begin{aligned} -\nabla^2 u + v \cdot \nabla u &= f \quad \text{on } \Omega \\ \frac{\partial u}{\partial \nu} + \varphi(u) &= 0 \quad \text{on } \Gamma \quad \text{and } u = 0 \text{ on } \partial\Omega \setminus \Gamma, \end{aligned} \tag{4.6}$$

where $v \in \mathbb{R}^2$, $\Gamma \subset \partial\Omega$, and $\varphi \colon \mathbb{R} \mapsto \mathbb{R}$ is a function of the form $\varphi(u) = \gamma \log(\beta + \alpha u)$, with $\alpha, \beta, \gamma > 0$. For our experiments we consider $\Omega = (0,1)^2$, $\Gamma = \{0\} \times [0,1]$, and

$v^T = [1, 1]$. We discretize (4.6) using finite differences with $n = 10000$ discretization points. The resulting convergence plots are presented in Figure 4.4, and they illustrate the convergence of synchronous and asynchronous NLRAS for (4.6) for $P = 16$ subdomains. We can clearly see that in this case the asynchronous method is more

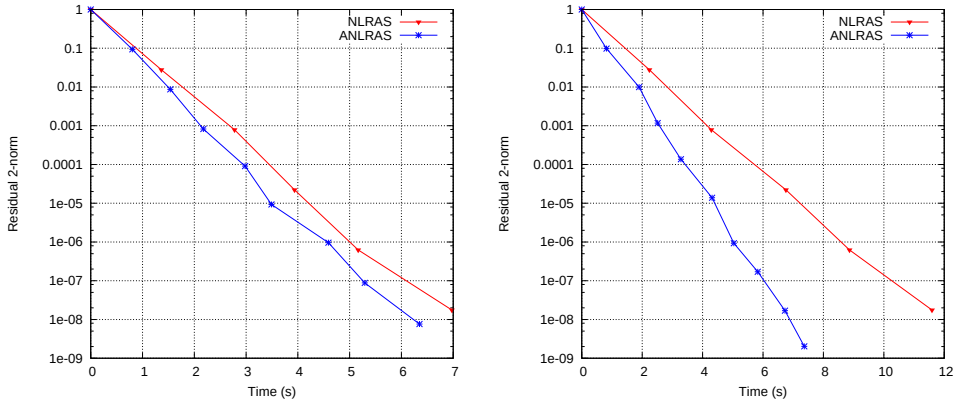

Fig. 4.4: Convergence curves of synchronous and asynchronous NLRAS for problem (4.6) with $\alpha = 1, \beta = 1, \gamma = 10^2$. Left: Homogeneous processor speed. Right: One processor is twice as slow.

advantageous, even in the case of an homogeneous network. This is due to the load imbalance among subdomains solves. Some subdomains (those lying in the boundary of Ω) require a nonlinear solver, while others require only a linear solver. The advantage is enhanced when one of the processors is twice as slow as the others.

5. Two-level nonlinear RAS (2L-NLRAS). In order to improve the convergence of NLRAS and ensure the scalablity of the method, an additional coarse correction can be incorporated. Since the problem is nonlinear, we need to use a nonlinear coarse grid correction. We opt here for the FAS (Full Approximation Scheme) method to perform the coarse grid correction; see, e.g., [9]. Let V_0 denote the coarse space and $R_0 : V \mapsto V_0$ the coarse restriction operator. We can define then by means of R_0 the coarse nonlinear problem $F_0 \colon V_0 \mapsto V_0$ which could be defined by using a coarse discretization of (2.1), or using a Galerkin approach, namely

$$F_0(u) = R_0 F(R_0^\top u).$$

The coarse correction $v_0 \in V_0$ using the Full Approximation Scheme is defined as the solution of the nonlinear problem

$$F_0(v_0 + R_0 u) = F_0(R_0 u) - R_0 F(u), \tag{5.1}$$

which can be incorporated into (2.7) either in an additive or multiplicative manner. Following [26], here we opt to use the additive variant. In order to prevent overcorrection on the overlap, we use a weighted two-level method of the form

$$u^{k+1} = u^k + \frac{1}{2}\sum_{i=1}^{P} R_{i,0}^\top v_{i,\delta}^k + \frac{1}{2} R_0^\top v_0^k; \tag{5.2}$$

see [24] for an alternative approach. The Full Approximation Scheme (FAS) [8] is a particular case of the nonlinear multilevel method (NMLM) of Hackbusch [28]. Although NMLM is a globally convergent iteration [28], there exist no such result for FAS. However, there exist cases for which the convergence of FAS was proven for a class of mildly nonlinear PDEs [33, Theorem 5.1].

In the rest of the section, we discuss the local convergence of the two-level NLRAS described in (5.2) and present numerical experiments which illustrate the benefits of the two-level approach.

Let us thus define the mapping $\tilde{T}$ corresponding to the fixed point iteration (5.2), i.e.,

$$\tilde{T}(u) = u + \frac{1}{2}\sum_{i=1}^{P} R_{i,0}^{\top} v_{i,\delta} + \frac{1}{2} R_0^{\top} v_0. \tag{5.3}$$

In order to show that (5.2) converges to the solution of (2.2), it suffices to show that there exists a norm such that $\|\tilde{T}'(u^*)\| < 1$. We have that

$$\tilde{T}'(u) = I + \frac{1}{2}\sum_{i=1}^{P} R_{i,0}^{\top} \frac{\partial v_{i,\delta}}{\partial u} + \frac{1}{2} R_0^{\top} \frac{\partial v_0}{\partial u}. \tag{5.4}$$

Moreover, we know that (4.1) holds. It remains to compute the derivative corresponding to the coarse correction v_0. Indeed, taking the derivative on both sides of (5.1) with respect to u, we obtain

$$\begin{aligned} F_0'(v_0 + R_0 u)\left(\frac{\partial v_0}{\partial u} + R_0\right) &= F_0'(R_0 u) R_0 - R_0 F'(u) \\ \frac{\partial v_0}{\partial u} + R_0 &= (F_0'(v_0 + R_0 u))^{-1}(F_0'(R_0 u) R_0 - R_0 F'(u)) \\ \frac{\partial v_0}{\partial u} &= (F_0'(v_0 + R_0 u)^{-1} F_0'(R_0 u) - I) R_0 - F_0'(v_0 + R_0 u)^{-1} R_0 F'(u). \end{aligned}$$

Substituting both terms $\dfrac{\partial v_{i,\delta}}{\partial u}$ and $\dfrac{\partial v_0}{\partial u}$ in (5.4), we obtain that the derivative of $\tilde{T}$ at $u = u^*$ reduces to

$$I - \frac{1}{2}\sum_{i=1}^{P} R_{i,0}^{\top} \left(R_{i,\delta} F'(u^*) R_{i,\delta}\right)^{-1} R_{i,\delta} F'(u^*) - \frac{1}{2} R_0^{\top} (R_0 F'(R_0^{\top} R_0 u^*) R_0^{\top})^{-1} R_0 F'(u^*). \tag{5.5}$$

In order to prove convergence of this two-level method, it suffices then to show that $\tilde{T}'(u^*)$ as in (5.5) is a contraction. We can use the same techniques as for the one-level method in Theorem 2.2. One added complication is that unlike, e.g., in [26], for our coarse grid restriction $R_0^{\top} R_0 \neq I$, and thus additional hypotheses are needed. Then, we can use arguments similar to those in the proof of [5, Theorem 7.3] to complete the proof, as we show below. Let

$$B = \sum_{i=1}^{P} R_{i,0}^{\top} (R_{i,\delta} F'(u^*) R_{i,\delta}^{\top})^{-1} R_{i,\delta}$$

and

$$B_0 = R_0^{\top} (R_0 F'(R_0^{\top} R_0 u^*) R_0^{\top})^{-1} R_0.$$

THEOREM 5.1. *Let u^* be the solution of (2.2). Assume that F' exists on a neighborhood of u^* and that $F'(u^*)$ and $F'(R_0^\top R_0 u^*)$ are nonsingular M-matrices. Assume further that there exists a weak regular splitting $F'(u^*) = M_0 - N_0$, and a diagonal matrix E_0 such that*

$$0 \leq E_0 \leq I,$$

and

$$B_0 = E_0 M_0^{-1}.$$

Let $w > 0$ be such that $F'(u^*)w > 0$. Then $\rho(\tilde{T}'(u^*)) < \|\tilde{T}'(u^*)\|_w < 1$, and the iteration (5.2) converges linearly to u^*.

Proof. We start by showing that $\tilde{T}' \geq 0$. Indeed, we observe that

$$\tilde{T}'(u^*) = \frac{1}{2}\underbrace{(I - BF'(u^*))}_{\geq 0} + \frac{1}{2}\underbrace{E_0(I - M_0^{-1}F'(u^*))}_{\geq 0} + \frac{1}{2}\underbrace{(I - E_0)}_{\geq 0}.$$

Now, let us re-write the local contraction operator at the root $T'(u^*)$ (5.5) as

$$\tilde{T}'(u^*) = I - \frac{1}{2}(B + B_0)F'(u^*).$$

We remark that B and B_0 are restrictions and prolongations of nonnegative matrices, where the restrictions and prolongation operators are nonnegative, they are nonnegative. Furthermore, $B \geq 0$ is nonsingular, in fact, it is a RAS preconditioner. Therefore, it cannot have a zero row, and thus $Be > 0$, for any positive vector e. Let $e = F'(u^*)w > 0$. Then, we have that $Be > 0$ and $B_0 e \geq 0$, so that

$$T'(u^*)w = w - \frac{1}{2}(Be + B_0 e) < w,$$

concluding that $\|T'(u^*)\|_w < 1$. ∎

We note that the hypotheses mentioned in Theorem (5.1) are not always satisfied. In particular, for our coarse grid, the nonnegativity of the operator $\tilde{T}'$ does not necessarily hold. For the particular case where the coarse grid corresponds to a boolean matrix, it is shown in [1] that the choice of $E_0 = R_0^\top R_0$ is possible.

We present now several numerical experiments comparing the performance of two-level NLRAS with the one-level counterpart for problem (4.4), and conduct scaling experiments by varying the number of subdomains.

We show in Figure 5.1 a weak scaling experiment for the problem (4.4). We compare the one- and two-level NLRAS for a 2×2 and 3×3 decomposition. The total number of degrees of freedom (DOFs) in each subdomain is kept the same for both cases, namely 625 variables per subdomain. As we can observe, the convergence of the one-level NLRAS deteriorates when the number of subdomains increases from $P = 4$ to $P = 9$. In contrast, the convergence rate of the two-level NLRAS remains approximately the same. We summarize weak scalability results with increasing numbers of subdomains for iteration (5.2) in Table 5.1, where we report number of iterations to reach the tolerance of 10^{-8}. These results show that the method without coarse correction fails to converge in 100 iterations, while for the two-level NLRAS, the number of iterations has a slow growth with the number of subdomains.

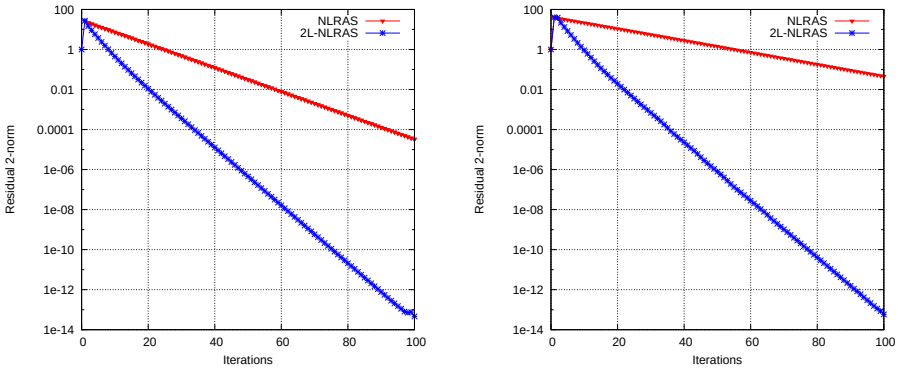

Fig. 5.1: Weak scaling experiment for one- and two-level NLRAS for problem (4.4) using a 2×2 checker-board partition (left), and 3×3 checker-board partition (right)

P	$\lvert V\rvert$	$\lvert V_0\rvert$	NLRAS	2L-NLRAS
4	2 500	36	160	62
9	5 625	81	323	64
16	10 000	144	554	94
25	15 625	225	853	85
36	22 500	324	1221	82
49	30 625	441	1657	80

Table 5.1: Number of iterations performed by NLRAS and two-level NLRAS for the problem (4.4) for different numbers of subdomains P and a tolerance of $\epsilon = 10^{-8}$

We repeat the same weak-scaling experiment but for a nonlinear diffusion equation problem of the form

$$\begin{aligned} -\nabla \cdot ((1+u^2)\nabla u) &= f(x,y), \quad \text{in } \Omega = (0,1)^2, \\ u(x,y) &= 0, \quad \text{on } \partial\Omega. \end{aligned} \tag{5.6}$$

We summarize in Table 5.2 the convergence results of iteration (5.2) for this problem. We point out that for a discretization of (5.6) using finite differences, $F'(u)$ is a nonsingular M-matrix for $u \in \Omega$, and therefore the hypothesis of our local convergence theorems are satisfied in this case. The results in Table 5.2 are similar to those in Table 5.1. The addition of the coarse correction makes the algorithm scalable in the sense that the number of iterations required to achieve the prescribed tolerance has a very slow growth as the number of subdomains increases. Comparing Tables 5.1 and 5.2, we note that for two problems with different type of nonlinearity, the two-level method behaves similarly in terms of number of iterations to convergence.

We end this section with two practical observations on the implementation of this two-level NLRAS method. The coarse correction step in (5.1) requires the use of Newton's iteration, which in turn necessitates the knowledge of the action of $F(u)$ as well as $F'(u)$. These quantities can be obtained by means of the local functions $F_{i,\delta}$

P	$\lvert V\rvert$	$\lvert V_0\rvert$	NLRAS	2L-NLRAS
4	2 500	36	183	63
9	5 625	81	373	64
16	10 000	144	642	97
25	15 625	225	992	89
36	22 500	324	1411	84
49	30 625	441	1913	85

Table 5.2: Number of iterations performed by NLRAS and two-level NLRAS for problem (5.6) for different number of subdomains P and a tolerance of $\epsilon = 10^{-8}$

as follows. By recalling the definition of $F_{i,\delta}$, we have that

$$F_0(u) = R_0 F(R_0^\top u) = R_0 \left(\sum_{i=1}^{P} R_{i,0}^\top F_{i,\delta}(R_0^\top u) \right).$$

Similarly, we obtain for the derivative of F_0

$$F_0'(u) = R_0 \left(\sum_{i=1}^{P} R_{i,0}^\top F_{i,\delta}'(R_0^\top u) \right) R_0^\top.$$

We comment on the appropriate choice of the coarse grid in this case to ensure convergence and scalability of the two-level nonlinear RAS. The one-level NLRAS has a residual $F(u)$, which is zero outside the overlap. Indeed, the global nonlinear residual is defined as $F(u^k)$, where u^k is the current approximation of the solution of (2.2). Moreover,

$$R_{i,\delta} F(u^k) = R_{i,\delta} F(u^{k-1} + \sum_{j=1}^{P} R_{j,0}^\top v_{j,\delta}^{k-1}) = F_{i,\delta}(u^{k-1} + R_{i,\delta}^\top v_{i,0}^{k-1} + \sum_{j \neq i} R_{j,0}^\top v_{j,\delta}^{k-1}).$$

Using the fact that $v_{i,\delta}^{k-1}$ satisfies $F_{i,\delta}(u^{k-1} + R_{i,0}^\top v_{i,\delta}^{k-1}) = 0$, it follows that the local nonlinear residual $R_{i,\delta} F(u^k)$ is zero outside the overlap. This shows that a good coarse grid should use this information for the construction of the coarse space. Thus, the coarse grid should be chosen on the overlap.

6. Asynchronous Two-Level NLRAS. In this section, we study the asynchronous version of the iteration (5.2). We present implementation details, local convergence theory, and numerical experiments. The information for the coarse grid correction comes from all subdomains, but it cannot be used until the information from all subdomains has arrived. Only then, the correction is added, and the process repeats. We implement this following ideas described in [26] for linear problems. We use a weighted additive NLRAS method (see [24] for multiplicative variants), and in order to prevent the coarse correction from over-correcting, we add the coarse grid only when all the subdomains have sent new data. This can be realized by defining a local boolean variable for each thread/subdomain. The coarse grid is then added when all these local boolean variables are set to true and this is done without a synchronization point. We describe the asynchronous two-level NLRAS in Algorithm 2.

We state and prove a result on the local convergence of two-level NLRAS as presented in Algorithm 2.

Algorithm 2 Asynchronous two-level nonlinear RAS (A2L-NLRAS)

Require: Initial vector u^0, tolerance ϵ.
Compute $\|F(u^0)\|_2$.
Set global_convergence=**false**, and local_convergence[k]=**false**, $k = 1, \ldots, P$.
Set local_update[k]=**false**, and local_correction[k]=**false**, $k = 1, \ldots, P$.
In parallel, each processor i:
while not global_convergence **do**
  if $i > 0$ **then**
    Compute $v_{i,\delta}$ by solving (2.6) using Newton's method.
    if local_correction[i] **then**
      Update $u = u + \frac{1}{2} R_{i,0}^\top v_{i,\delta} + \frac{1}{2} R_{i,0}^\top R_{i,\delta} v_0$
      Set local_correction[i]=**false**
    else
      Update $u = u + R_{i,0}^\top v_{i,\delta}$
      Set local_update[i]=**true**
    end if
    if $\|F_{i,0}(u)\|_2 / \|F(u^0)\|_2 < \epsilon/\sqrt{P}$ **then**
      Set local_convergence[i]=**true**
    end if
    if i==1 **then**
      if local_convergence[k], $\forall k = 1, \ldots, P$ **then**
        Set global_convergence=**true**
      end if
    end if
  else
    if local_update[k], $\forall k = 1, \ldots, P$ **then**
      Compute the coarse correction v_0 using (5.1).
      Set local_correction[k]=**true**, $k = 1, \ldots, P$
      Set local_update[k]=**false**, $k = 1, \ldots, P$
    end if
  end if
end while (**for processor** i)
Output: u^*.

THEOREM 6.1. *Let u^* be the solution of (2.2). Assume that F' exists on a neighborhood of u^* and that $F'(u^*)$ and $F'(R_0^\top R_0 u^*)$ are nonsingular M-matrices. Let $w > 0$ be such that $T'(u^*)w > 0$. Assume that the conditions (i)–(iii) hold. Then Algorithm 2 converges to u^*, the solution of (2.2).*

Proof. We will use Theorem 3.3. To that end, we define a sequence of operators T^k, which is either

$$T^k(u) = \tilde{T}(u) = u + \frac{1}{2} \sum_{i=1}^{P} R_{i,0}^\top v_{i,\delta} + \frac{1}{2} R_0^\top v_0, \tag{6.1}$$

if the coarse grid in Algorithm 2 is added, or

$$T^k(u) = T(u) = u + \sum_{i=1}^{P} R_{i,0}^\top v_{i,\delta}, \tag{6.2}$$

otherwise. Of course the operators (6.1) are those of the two-level NLRAS method (5.3), while those in (6.2) are those of the one-level method (2.8). It follows that both opera-

tors have u^* as a common fixed point, and in both cases, $\|T^k\|_w < 1$, by Theorems 2.2 and 5.1. Thus, from Theorem 3.3, Algorithm 2 converges to u^*. □

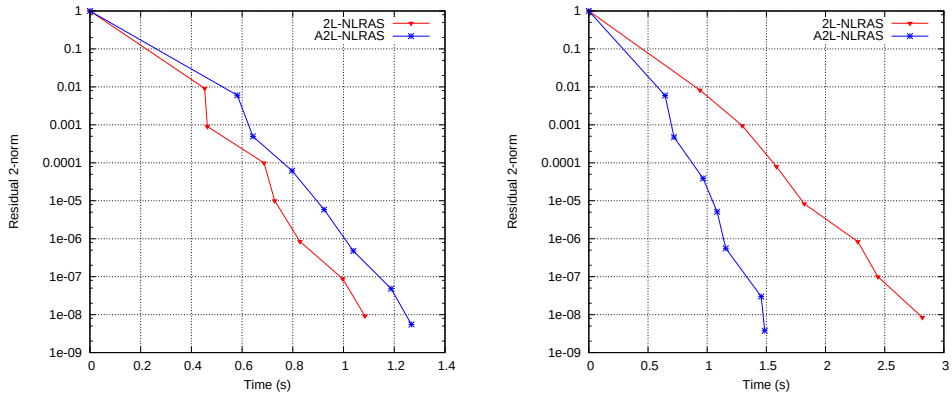

Fig. 6.1: Convergence curves of synchronous and asynchronous two-level NLRAS for problem (4.4) with $P = 16$. Left: Homogeneous network. Right: One processor four times slower than the others. .

We compare numerically the parallel synchronous and asynchronous two-level NLRAS for the nonlinear problem (4.4). We keep the same setting used in Section 4. In particular, the Jacobians are computed exactly for the inner Newton's solves. The coarse grid is chosen using a full weighting restriction matrix. We use 10 000 discretization points for the domain $\Omega = (0, 1)^2$ which we partition into 16 subdomains. The total size of the coarse grid is 144 discretization points. We test our methods in two settings: first with all processors of the same speed, and then, with one processor slower than the others, in this case, four times slower. These experiments are reported in Figure 6.1. We can clearly see that in the case of a inhomogeneous network the asynchronous two-level method is faster than its synchronous counterpart. For our weak scaling study, varying the number of subdomains, we report in Figure 6.2 the time required to reach a tolerance of $\epsilon = 10^{-8}$ for the synchronous and asynchronous two-level NLRAS for problems (4.4) and (5.6). We keep the computation unbalanced where one processor is four times slower than the others. We can clearly see that the asynchronous method converges faster than the synchronous one.

We conclude with a table comparing execution times and the number of processor updates for the one-level and two-level methods in both the synchronous and asynchronous versions for problem (4.4) in 2D and 3D using $P + 1$ processors. We note that each subdomain is assigned to a processor and the coarse grid is computed on a separate one. In the 3D case, the domain $\Omega = (0, 1)^3$ is decomposed into P equally sized bricks, and their number is the same in each direction. The size of each local problem is 1000×1000. We assume that one processor is four times slower than the others. The convergence results are reported in Table 6.1. We see from Table 6.1 that the asynchronous method outperforms the synchronous NLRAS. The addition of the coarse grid correction makes both methods even faster, with an advantage for

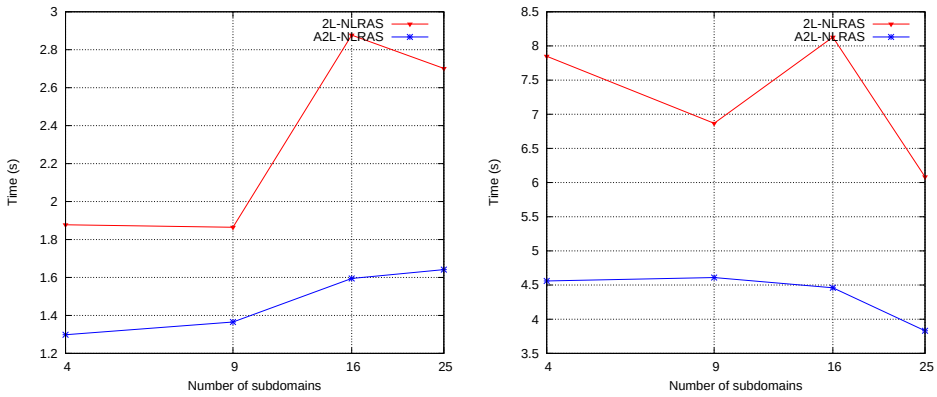

Fig. 6.2: Convergence time in seconds of synchronous and asynchronous two-level NLRAS versus the number of subdomains for $\epsilon = 10^{-8}$ where one processor is four times slower than the other ones. Left: Problem (4.4). Right: Problem (5.6).

the asynchronous one. The results in Table 6.1 also shows that the timing required for the coarse step increases for increasing number of subdomains.

Observe that as expected, the asynchronous method has a larger number of updates on average than the number of iterations of the synchronous method. Nevertheless, its execution time is faster, confirming that it can outperform the synchronous method. This is true for both the one- and two-level methods.

P	$\|V\|$	$\|V_0\|$	NLRAS				2L-NLRAS					
			sync		async		sync			async		
			time	#iter	time	#updates	time	coarse	#iter	time	coarse	#updates
4	2 500	36	4.44	160	3.38	374	1.71	0.20	62	1.39	0.57	131
9	6 525	81	9.67	323	5.22	581	1.87	0.45	64	1.40	0.72	153
16	10 000	144	16.42	554	7.18	880	2.86	1.30	94	1.47	0.96	170
25	15 625	225	26.15	853	10.98	1339	2.82	1.66	85	1.68	1.25	187
8	8 000	125	21.53	72	17.28	163	17.64	1.68	56	14.17	4.02	150
27	27 000	343	47.51	147	30.71	268	24.23	9.55	77	17.71	5.82	159
64	64 000	1000	56.73	251	41.99	447	17.86	17.71	67	16.70	13.51	182

Table 6.1: Time (in sec) and the number of iterations or average number of updates for synchronous and asynchronous NLRAS and its two-level variant for problem (4.4) in 2D (top) and 3D (bottom) to reach a tolerance $\epsilon = 10^{-8}$ for different numbers of subdomains with constant local problem size.

7. Conclusion. In summary, the new parallel nonlinear RAS algorithm converges to the unique solution of the problem in the original domain of the PDE. An asynchronous version can be about than twice as fast or faster, depending on the inhomogeneities of the computation, and the communication costs. The two-level

version can be an order of magnitude faster, and its asynchronous version faster still, exhibiting slow growth in execution times when the number of processors increases in weak scaling.

REFERENCES

[1] Andreas Frommer and Daniel B. Szyld. Weighted Max Norms, Splittings, and Overlapping Additive Schwarz Iterations. *Numerische Mathematik*, 83:259–278, 1999.

[2] Jacques Mohcine Bahi, Sylvain Contassot-Vivier, and Raphaël Couturier. *Parallel Iterative Algorithms: from Sequential to Grid Computing*. CRC Press, 2007.

[3] Zhong-Zhi Bai, Violeta Migallón, José Penadés, and Daniel B. Szyld. Block and asynchronous two-stage methods for mildly nonlinear systems. *Numerische Mathematik*, 82:1–20, 1999.

[4] Gérard M. Baudet. Asynchronous iterative methods for multiprocessors. *Journal of the Association for Computing Machinery*, 25:226–244, 1978.

[5] Michele Benzi, Andreas Frommer, Reinhard Nabben, and Daniel B. Szyld. Algebraic theory of multiplicative Schwarz methods. *Numerische Mathematik*, 89(4):605–639, 2001.

[6] Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[7] Amit Bhaya, Eugenius Kaszurewicz, and Francisco Mota. Asynchronous block-iterative methods for almost linear equations. *Linear Algebra and its Applications*, 154:487–508, 1991.

[8] Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[9] Achi Brandt. Guide to multigrid development. In Wolfgang Hackbusch and Ulrich Trottenberg, editors, *Multigrid methods*, pages 220–312. Springer, 1982.

[10] Xiao-Chuan Cai and David E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. *SIAM Journal on Scientific Computing*, 24(1):183–200, 2002.

[11] Xiao-Chuan Cai and Xuefeng Li. Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity. *SIAM Journal on Scientific Computing*, 33(2):746–762, 2011.

[12] Xiao-Chuan Cai and Marcus Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21(2):792–797, 1999.

[13] Faycal Chaouqui, Martin J. Gander, Pratik M. Kumbhar, and Tommaso Vanzan. Linear and nonlinear substructured Restricted Additive Schwarz iterations and preconditioning. *Numerical algorithms*, inprint, 2022.

[14] Ming Chau, Didier El Baz, Ronan Guivarch, and Pierre Spiteri. MPI implementation of parallel subdomain methods for linear and nonlinear convection–diffusion problems. *Journal of Parallel and Distributed Computing*, 67(5):581–591, 2007.

[15] Gabriele Ciaramella and Tommaso Vanzan. Substructured two-grid and multi-grid domain decomposition methods. *Numerical Algorithms*, pages 1–36, 2022.

[16] Ljubomir B. Ciric. Generalized contractions and fixed-point theorems. *Publications de l'Institut Mathématique (Beograd)(NS)*, 12(26):19–26, 1971.

[17] Victorita Dolean, Martin J. Gander, Walid Kheriji, Felix Kwok, and Roland Masson. Nonlinear preconditioning: How to use a nonlinear Schwarz method to precondition Newton's method. *SIAM Journal on Scientific Computing*, 38(6):A3357–A3380, 2016.

[18] Maksymilian Dryja and Wolfgang Hackbusch. On the nonlinear domain decomposition method. *BIT Numerical Mathematics*, 37(2):296–311, 1997.

[19] Evridiki Efstathiou and Martin J. Gander. Why restricted additive Schwarz converges faster than additive Schwarz. *BIT Numerical Mathematics*, 43(5):945–959, 2003.

[20] Mouhamed Nabih El Tarazi. Some convergence results for asynchronous algorithms. *Numerische Mathematik*, 39(3):325–340, 1982.

[21] Andreas Frommer and Hartmut Schwandt. Asynchronous parallel methods for enclosing solutions of nonlinear equations. *Journal of Computational and Applied Mathematics*, 60(1-2):47–62, 1995.

[22] Andreas Frommer and Daniel B Szyld. On asynchronous iterations. *Journal of Computational and Applied Mathematics*, 123(1-2):201–216, 2000.

[23] Andreas Frommer and Daniel B. Szyld. An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM Journal on Numerical Analysis*, 39:463–479, 2001.

[24] Guillaume Gbikpi Benissan and Frédéric Magoulès. Asynchronous Multiplicative Coarse-Space Correction. *SIAM Journal on Scientific Computing*, 44(3):C237–C259, 2022.

[25] Luc Giraud and Pierre Spiteri. Résolution parallèle de problèmes aux limites non linéaires. *ESAIM: Mathematical Modelling and Numerical Analysis*, 25(5):579–606, 1991.

[26] Christian Glusa, Erik G. Boman, Edmond Chow, Sivasankaran Rajamanickam, and Daniel B. Szyld. Scalable Asynchronous Domain Decomposition Solvers. *SIAM Journal on Scientific Computing*, pages C384–C409, 2020.

[27] Gaël Guennebaud, Benoit Jacob, et al. Eigen. *URl: http://eigen. tuxfamily. org*, 2010.

[28] Wolfgang Hackbusch. *Multigrid Methods and Applications*, volume 4 of *Springer Series in Computational Mathematics*. Srpinger, Berlin Heidelberg, 1985.

[29] Alexander Heinlein and Martin Lanser. Additive and Hybrid Nonlinear Two-Level Schwarz Methods and Energy Minimizing Coarse Spaces for Unstructured Grids. *SIAM Journal on Scientific Computing*, 42(4):A2461–A2488, 2020.

[30] Feng-Nan Hwang and Xiao-Chuan Cai. Improving Robustness and Parallel Scalability of Newton Method Through Nonlinear Preconditioning. In Ralf Kornhuber, Ronald Hoppe, Jacques Périaux, Olivier Pironneau, Olof Widlund, and Jinchao Xu, editors, *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 201–208, Berlin, Heidelberg, 2005. Springer.

[31] Olga A. Ladyzhenskaya and Nina N. Ural'tseva. Linear and Quasilinear Elliptic Equations. *Academic Press, New York*, 1968.

[32] Jean-Claude Miellou, Didier El Baz, and Pierre Spiteri. A new class of asynchronous iterative algorithms with order intervals. *Mathematics of Computation*, 67(221):237–255, 1998.

[33] Arnold Reusken. Convergence of the multigrid full approximation scheme for a class of elliptic mildly nonlinear boundary value problems. *Numerische Mathematik*, 52(3):251–277, 1987.

[34] Pierre Spiteri. Parallel asynchronous algorithms: A survey. *Advances in Engineering Software*, 149:102896, 2020.

[35] Pierre Spiteri, Jean-Claude Miellou, and Didier El Baz. Parallel asynchronous Schwarz and multisplitting methods for a nonlinear diffusion problem. *Numerical Algorithms*, 33(1-4):461–474, 2003.

[36] Daniel B. Szyld. Different models of parallel asynchronous iterations with overlapping blocks. *Computational and Applied Mathematics*, 17:101–115, 1998.

[37] Jian-Jun Xu. Convergence of partially asynchronous block quasi-Newton methods for nonlinear systems of equations. *Journal of Computational and Applied Mathematics*, 103(2):307–321, 1999.