Numer. Math.
DOI 10.1007/s00211-017-0872-z

Numerische Mathematik

Asynchronous optimized Schwarz methods with and without overlap

Frédéric Magoulès[1] · **Daniel B. Szyld**[2] · **Cédric Venet**[1]

Received: 29 June 2015 / Revised: 21 November 2016

Abstract An asynchronous version of the optimized Schwarz method for the solution of differential equations on a parallel computational environment is studied. In a one-way subdivision of the computational domain, with and without overlap, the method is shown to converge when the optimal artificial interface conditions are used. Convergence is also proved for the Laplacian operator under very mild conditions on the size of the subdomains, when approximate (non-optimal) interface conditions are utilized. Numerical results are presented on a large three-dimensional problem on modern parallel clusters and supercomputers illustrating the efficiency of the asynchronous approach.

Mathematics Subject Classification 65F10 · 65N22 · 15A06

The first and third authors acknowledge partial financial support from the CRESTA project (European Commission). The second author's work was supported in part by the U.S. National Science Foundation under Grant DMS-1418882 and the U.S. Department of Energy under Grant DE-SC0016578. This version of 10 February 2017.

✉ Daniel B. Szyld
szyld@temple.edu

Frédéric Magoulès
frederic.magoules@hotmail.com

[1] CentraleSupelec, Université Paris-Saclay, Grande Voie des Vignes, 92295 Châtenay Malabry Cedex, France

[2] Department of Mathematics, Temple University (038-16), 1805 N. Broad Street, Philadelphia, PA 19122-6094, USA

Published online: 08 March 2017

1 Introduction

Schwarz iterative methods were originally devised to show existence of solutions of elliptic problems on irregular domains [35], and were revived as numerical methods in the 1980s; see, e.g., the pioneering paper [19], or the excellent books [30,34,36,37], and the many references therein.[1] For these Schwarz methods, one may consider the solution of a general problem of the form

$$\begin{cases} \mathcal{L}(u) = f & \text{in } \Omega \\ \mathcal{C}(u) = g & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where $\mathcal{L}$ and $\mathcal{C}$ are partial differential operators defined on the domain $\Omega \subseteq \mathbb{R}^d$ and its boundary, respectively. This domain is (artificially) split into two ($p = 2$) or more (possibly overlapping) subdomains, i.e., we have $\Omega = \cup_{s=1,\ldots,p} \Omega^{(s)}$. In essence one is introducing new artificial boundary conditions on the interfaces between these subdomains. In the classical formulation, these artificial boundary conditions are of Dirichlet type. Given an initial approximation $u(0)$, the method progresses by solving for $u(n+1)$ at iteration $n+1$ the Eq. (1) restricted to each subdomain $\Omega^{(s)}$ using as boundary data on $\partial\Omega^{(s)} \backslash \partial\Omega$ the values for $u(n)$ at iteration n. This procedure is inherently parallel, since the (approximate) solutions on each subdomain can be performed by a different processor.

Convergence of these iterations can be guaranteed under mild conditions, but it is in general rather slow, comparable to the Block Jacobi or Block Gauss–Seidel methods for linear algebraic systems. Faster convergence can be achieved by using Robin and mixed boundary conditions on the interfaces. These boundary conditions are, for example, of the form $\frac{\partial u}{\partial \nu} + \alpha u$ or $\frac{\partial u}{\partial \nu} + \alpha u + \beta \frac{\partial^2 u}{\partial \tau^2}$ (often called OO0 and OO2 conditions, respectively), where ν and τ refer to the normal and tangential directions, respectively. In this way one can optimize the Robin parameter(s) and obtain a very fast method. This technique has been termed optimized Schwarz method, and derived for several situations including Maxwell equations [6,8], Helmholtz equations [13,16], fluid dynamics [7], convection diffusion [29], reaction diffusion problems [15], and heterogeneous media [20,21]. See also the review papers [14,31], and references therein as well as Sect. 3 for more details, and [17,26,27], for an algebraic version of this optimized approach.

Asynchronous methods refer to parallel iterative procedures where each process performs its task without waiting for other processes to be completed, i.e., with whatever information it has locally available and with no synchronizations with other processes. Asynchronous methods can be faster than their synchronous counterparts precisely because each processor does not need to wait for information from the other processors to proceed. Mathematical models of this computational paradigm were developed in the 1980s and 1990s and convergence proofs given; see, e.g., the survey

[1] For proceedings of the frequent conferences on this topic and for an extensive bibliography see the website www.ddm.org.

[12] and references therein, as well as Sect. 2 below, where some of these models and convergence results are reviewed.

Asynchronous (classical) Schwarz methods were successfully explored, e.g., in [5,11,18]. Recently, the first author proposed the use of asynchronous methods in the context of (the much faster) optimized Schwarz iterations [22], and showed numerical experiments for the solution of an exterior Helmholtz problem obtaining gains in computational time in the 30–50% range over the synchronous version on an heterogeneous cluster composed of 64 processors.

In this paper, we provide proofs of the convergence of asynchronous optimal and optimized Schwarz methods (with and without overlap[2]) in two instances. First, in Sect. 3 we prove convergence of the asynchronous iterations for a general partial differential operator and optimal interface conditions. Then, in Sect. 4 we show convergence for approximate optimized conditions for the Laplacian operator on $\mathbb{R}^2$. In a companion paper, we extend these results to the Helmholtz equation [28]. Our theorems are very general in the sense that they can be applied to several artificial interface conditions, including OO0 and OO2. We then illustrate the applicability of the method with numerical experiments of the gravitational analysis of the Chicxulub crater in the Yucatan peninsula, with a mesh of almost 147 million unknowns. Our experiments were ran on two different architectures. One, which is more heterogeneous, is a cluster with 160 cores, and the other is a supercomputer composed of 1068 computing nodes.

Part of our contribution is to present a proof of convergence of optimized Schwarz methods for several subdomains ($p > 2$), to use this result to provide the first proof of convergence of asynchronous optimized Schwarz methods, and to report numerical experiments on real data with a very large number of degrees of freedom.

We believe that the asynchronous optimized Schwarz methods discussed in this paper have the potential to be extremely useful in high performance computing environments. The main reason for this statement is precisely that no processor needs to wait for information coming from the other processors. The new theory developed here allows us to combine this asynchronous paradigm with a very fast method (optimized Schwarz), producing an extremely effective and robust computational tool.

The paper is structured as follows. We begin by giving a very general description of synchronous iterative methods for fixed point problems. This sets the stage for a review of convergence results of asynchronous methods for these problems. Our proof of convergence of the asynchronous Schwarz iterations then proceeds with the use of functional analysis, Fourier transform, and linear algebra techniques to show that one can satisfy the hypotheses, appropriately interpreted, of the convergence results for general asynchronous methods. Finally the numerical experiments illustrates the robustness, performance and efficiency of the proposed method on parallel architectures.

[2] The phrase "without overlap" is used in the domain decomposition literature to mean that the subdomains share only the boundary, and no interior points; see further the comment in Remark 1.

2 Mathematical description of general synchronous and asynchronous parallel iterations

2.1 Classical synchronous iterations

We first consider a general fixed point problem of the form $x = \mathcal{T}x$ defined on the set X, with $\mathcal{T} : X \mapsto X$ and with a unique solution x^*. A stationary iterative algorithm is the successive application of the function $\mathcal{T}$ to an initial approximation of the solution. Thus, the mathematical formulation of the iterative algorithm, given an initial approximation $x(0)$ is

$$x(n+1) = \mathcal{T}(x(n)) \quad \text{for } n = 0, 1, \ldots \tag{2}$$

The result $x(n+1)$ of each iteration is an approximation of the solution x^*. The algorithm terminates when a sufficiently precise approximation of the solution has been found. The method used to determine the precision of the approximation is algorithm and problem dependent. An algorithm is said to converge for a given problem if it converges for any initial approximation $x(0)$, i.e., if

$$\text{for every } x(0) \in X, \quad \lim_{n\to\infty} x(n) = x^*.$$

A parallel iterative algorithm is an iterative algorithm where the application of the function $\mathcal{T}$ is performed in parallel. Suppose that there are p processes and let us define for each process s a set $X^{(s)}$ on which the process works. While $X^{(s)}$ is often a subset of X, this is not necessary. The parallel algorithm could work on artificial variables not present in the original problem, such as Lagrange multipliers. This formulation also allows one to handle overlap, or the same variable appearing in more than one set $X^{(s)}$. A mapping from the Cartesian product of the local spaces

$$P : X^{(1)} \times \cdots \times X^{(p)} \mapsto X$$

is defined which allows to extract the variables of the original problem from those in the parallel algorithm, and in the case of overlap, to choose the appropriate representative for that variable (or its computation). Each process has its own function $\mathcal{T}^{(s)}$ with

$$\mathcal{T}^{(s)} : X^{(1)} \times \cdots \times X^{(p)} \mapsto X^{(s)}.$$

The synchronous parallel iteration is defined for each process s by starting with initial approximations $x^{(s)}(0)$, $s = 1, \ldots, p$, and computing

$$x^{(s)}(n+1) = \mathcal{T}^{(s)}\left(x^{(1)}(n), \ldots, x^{(p)}(n)\right) \quad \text{for } s = 1, \ldots, p, \quad n = 0, 1, \ldots, \tag{3}$$

with $x^{(s)} \in X^{(s)}$. Then, the approximation of the solution at the iteration $n+1$ is given by

$$x(n+1) = P\left(x^{(1)}(n+1), \ldots, x^{(p)}(n+1)\right).$$

This formulation is clearly parallel. At each iteration, each process s has a function to apply and then sends the result to the other processes. The functions have no dependencies, so communication between processes is only needed between iterations. In this model, the general case where each process needs the result from all the others is used. In a practical implementation, data transfers can be omitted when there are no data dependencies between the processes, i.e., if the function $\mathcal{T}^{(s)}$ does not depend on its q-th argument, there is no need for the process q to send its data to process s.

A sequential version equivalent to this parallel algorithm can be derived as follows. Let us define the space $\widetilde{X} = X^{(1)} \times \cdots \times X^{(p)}$ and an operator $\widetilde{\mathcal{T}}(n) : \widetilde{X} \mapsto \widetilde{X}$ by

$$\begin{aligned}\tilde{x} &= \left(x^{(1)}, \ldots, x^{(p)}\right) \mapsto \widetilde{\mathcal{T}}(\tilde{x}) \\ &= \left(\mathcal{T}^{(1)}\left(x^{(1)}, \ldots, x^{(p)}\right), \ldots, \mathcal{T}^{(p)}\left(x^{(1)}, \ldots, x^{(p)}\right)\right).\end{aligned}$$

Then, with this operator $\widetilde{\mathcal{T}}$ (over the space $\widetilde{X}$) we can define iterations of the form (2), and this sequential iterative algorithm is equivalent to the synchronous parallel algorithm defined by (3) in the sense that every iteration n of one algorithm will be identical to that of the other algorithm.

Generally, the extended set $\widetilde{X}$ is not equal to the set X of the original problem. Some variables may be duplicated (overlapping) or state variables added. Some algorithms can even work on transformed variables (Lagrange multipliers). The function P must be surjective (note that this implies that $\dim \widetilde{X} \geq \dim X$) so that every possible value of the problem domain can be represented but this is the only constraint on the space $\widetilde{X}$. Sometimes $\widetilde{X} = X$, in this case P is just the identity (at least from a mathematical point of view, in practice, the variables would probably have been renumbered). The sequential equivalent version is useful since one can then prove convergence of the algorithm given by (2), and this would automatically imply the convergence of the parallel counterpart (3). We note though that this is only true for synchronous iterations. Nonetheless, this construct is also useful when analyzing asynchronous iterations, as we do in the next section.

We end this section recalling an important convergence theorem for fixed point problems (which are the focus of this paper); cf., e.g., [33, Theo. 10.1.2]. We do this to illustrate the similarity in structure to the results on asynchronous methods in the next section.

Theorem 1 *Consider a stationary iteration with operator $\mathcal{T}$ with fixed point x^* and initial approximation $x(0)$. Assume that there is a norm for which $\mathcal{T}$ is a contraction, i.e., for which $\|\mathcal{T}\| < 1$, then the algorithm is convergent. Moreover, if $\mathcal{T}$ is Lipshitz with constant α, then the approximation error at the iteration n satisfies*

$$\left\|x(n) - x^*\right\| \leq \alpha^n \left\|x(0) - x^*\right\|.$$

2.2 Asynchronous iterations

Asynchronous iterations are iterations where an operation of the type (3) is performed, and once it is finished, it is repeated with whatever new information has arrived, but

without synchronization, i.e., without waiting to the other processors to complete their task. Note that in this setting, the concept of iteration counter n loses its meaning, and one can think of simply a new approximation to x^* computed with whatever information is available from the other processes. The latter may have been just computed, or possibly computed much earlier.

Not every parallel iterative algorithm can be used asynchronously and expected to converge. On the other hand, all asynchronous iterative algorithms may be used as synchronous iterations, since they correspond to a specific execution of asynchronous iterations. Some of the classical synchronous iterative methods can be used with asynchronous iterations without change. However, their convergence requirements are usually stricter than their synchronous counterpart. We present below the mathematical framework used to describe and study these algorithms. We follow the model introduced by Bertsekas [3,4].

For a mathematical model of these asynchronous iterations on p processors, let us denote by $\{\sigma(n)\}_{n\in\mathbb{N}}$ the sequence of non-empty subsets of $\{1,\ldots,p\}$, defining which processes update their components at the "iteration" n, where here "iteration" can be thought of a time stamp. We call these, sets of update indexes. Define further for $s, q \in \{1,\ldots,p\}$, $\left\{\tau_q^{(s)}(n)\right\}_{n\in\mathbb{N}}$ a sequence of integers, where $\tau_q^{(s)}(n)$ represents the iteration number (or time stamp) of the data coming from process q and available on process s at time n. Thus, these are the time stamps of previous computations that are used by process s, and thus, the quantities $n-\tau_q^{(s)}(n)$ are sometimes called *delays*. We can now define, for each process s, the asynchronous iterations as follows.

$$\begin{cases} x^{(s)}(0) = x_0^{(s)} \\ x^{(s)}(n+1) = \begin{cases} \mathcal{T}^{(s)}\left(x^{(1)}(\tau_1^{(s)}(n)),\ldots,x^{(p)}(\tau_p^{(s)}(n))\right) & \text{if } s\in\sigma(n) \\ x^{(s)}(n) & \text{if } s\notin\sigma(n)\,. \end{cases} \end{cases} \tag{4}$$

In other words, at the time $n+1$, either $x^{(s)}$ is not updated (if $s\notin\sigma(n)$) or it is updated with the result of applying the (local) operator $\mathcal{T}^{(s)}$ to the variables computed at times $\tau_i^{(s)}$. We further assume that the three following conditions are satisfied

$$\forall s,q\in\{1,\ldots,p\},\forall n\in\mathbb{N}^*,\tau_q^{(s)}(n)\leq n, \tag{5}$$

$$\forall s\in\{1,\ldots,p\},\text{card}\left\{n\in\mathbb{N}^*|s\in\sigma(n)\right\}=+\infty, \tag{6}$$

$$\forall s,q\in\{1,\ldots,p\},\lim_{n\to+\infty}\tau_q^{(s)}(n)=+\infty. \tag{7}$$

Condition (5) indicates that data used at the time n must have been produced before time n, i.e., time does not flow backward. Condition (6) means that no process will ever stop updating its components. Condition (7) corresponds to the fact that new data will always be provided to the process. In other words, no process will have a piece of data that is never updated.

The following notation is reused in this asynchronous context:

$$\widetilde{X}=X^{(1)}\times\cdots\times X^{(p)},\tilde{x}=\left(x^{(1)},\ldots,x^{(p)}\right)\quad\text{and}\quad\widetilde{\mathcal{T}}=\left(\mathcal{T}^{(1)},\mathcal{T}^{(2)},\cdots,\mathcal{T}^{(p)}\right).$$

We use this notation in the following assumption [3, §6.2].

Assumption 2 There is a sequence of non-empty sets $\left\{\widetilde{X}(n)\right\}$ with

$$\cdots \subset \widetilde{X}(n+1) \subset \widetilde{X}(n) \subset \cdots \subset \widetilde{X}$$

satisfying the following two conditions:

(**a**) Synchronous Convergence Condition.
For every $n \in \mathbb{N}^*$ and $\tilde{x} \in \widetilde{X}(n)$, $\widetilde{T}(\tilde{x}) \in \widetilde{X}(n+1)$. Furthermore, if $\{y(n)\}$ is a sequence such that $y(n) \in \widetilde{X}(n)$ for every n, then every limit point of $\{y(n)\}$ is a fixed point of $\widetilde{T}$.
(**b**) Box Condition. For every n, there exist sets $X^{(s)}(n) \subset X^{(s)}$ such that

$$\widetilde{X}(n) = X^{(1)}(n) \times X^{(2)}(n) \times \cdots \times X^{(p)}(n).$$

We are ready to present a convergence theorem for asynchronous iterative algorithms, whose proof can be found in [4]; see also [38]. We mention that both the box condition, by which every set is a Cartesian product of the local spaces, and the strict containment of those sets are fundamental in the proof of this theorem.

Theorem 3 *Let Assumption 2 and conditions (5)–(7) hold. Then, for any initial vector*

$$\tilde{x}(0) = \left(x(0)^{(1)}, \ldots, x(0)^{(p)}\right) \in \widetilde{X}(0),$$

the limit points of the sequence $(\tilde{x}(n))_n$ *generated by an asynchronous iterative algorithm of the form (4) are fixed points of* $\widetilde{T}$.

Note that if $\widetilde{T}$ has a unique fixed point, then the asynchronous algorithm (4) satisfying the hypotheses of Theorem 3 converges to this unique fixed point.

We end this section with a result due to El Tarazi [10], specifically for the case that there exists a unique fixed point, i.e., if there is $\tilde{x}^*$ such that $\widetilde{T}(\tilde{x}^*) = \tilde{x}^*$. To that end, we define for a positive vector $w^T = (w_1, \ldots, w_N)$, i.e., one in which each component $w_i > 0$, the weighted maximum norm as

$$\|x\|_\infty^w = \max_{1 \le i \le N} \frac{|x_i|}{w_i}.$$

The following result is due to El Tarazi [10].

Theorem 4 *Let* $\widetilde{T}$ *be a contraction mapping with respect to a weighted maximum norm* $\|.\|_\infty^w$ *with a unique fixed point and assume that the conditions (5)–(7) hold. Then, the asynchronous iterative algorithm of the form (4) converges to the unique fixed point of* $\widetilde{T}$.

3 Schwarz method with optimal interface conditions

We study a very general problem of the form (1), where $\mathcal{L}$ and $\mathcal{C}$ are partial differential operators, and Ω a bounded Lipschitz domain. The domain Ω is decomposed in a one-way splitting into p (possibly overlapping) subdomains $\Omega^{(s)}$, $1 \leq s \leq p$, in such a way that each subdomain has two (or one) neighbouring subdomains. For each subdomain $\Omega^{(s)}$ the following sets are defined:

$$\Omega^{(s-)} = \bigcup_{q<s} \Omega^{(q)} \backslash \Omega^{(s)} \quad \text{and} \quad \Omega^{(s+)} = \bigcup_{q>s} \Omega^{(q)} \backslash \Omega^{(s)},$$

as illustrated in Fig. 1. The subdomains are numbered so that $\Omega^{(s)} \cap \Omega^{(q)} \neq \emptyset$ if and only if $|s-q| \leq 1$. As a consequence we have $\Omega^{(s-)} \cup \Omega^{(s)} \cup \Omega^{(s+)} = \Omega$ and $\Omega^{(s-)} \cap \Omega^{(s+)} = \emptyset$. The (artificial) interface of $\Omega^{(s)}$, $\partial\Omega^{(s)} \backslash \partial\Omega$, is split into two disjoint interfaces $\Gamma_l^{(s)}$ and $\Gamma_r^{(s)}$: $\Gamma_l^{(s)} = \Omega^{(s-)} \cap \partial\Omega^{(s)}$ on the left and $\Gamma_r^{(s)} = \Omega^{(s)} \cap \partial\Omega^{(s+)}$ on the right. On the leftmost subdomain ($s = 1$), $\Omega^{(s-)}$ and $\Gamma_l^{(s)}$ are reduced to $\Omega^{(s-)} = \emptyset$ and $\Gamma_l^{(s)} = \emptyset$ respectively. Similarly, on the rightmost subdomain ($s = p$), the sets are reduced to $\Omega^{(s+)} = \emptyset$ and $\Gamma_r^{(s)} = \emptyset$. The values f and g can be included in the differential operators $\mathcal{L}$ and $\mathcal{C}$ respectively, and thus, without loss of generality we assume from now on that $f = 0$ and $g = 0$.

We define the Steklov–Poincaré operators $\Lambda^{(s-)} : \Gamma_l^{(s)} \mapsto \Gamma_l^{(s)}$ such that $\Lambda^{(s-)}(w) = \frac{\partial v}{\partial v_l^{(s)}}$, where v is the solution of the system

$$\begin{cases} \mathcal{L}(v) = 0 & \text{in } \Omega^{(s-)}, \\ v = w & \text{on } \Gamma_l^{(s)}, \\ \mathcal{C}(v) = 0 & \text{on } \partial\Omega \cap \partial\Omega^{(s-)}, \end{cases}$$

and $\Lambda^{(s+)} : \Gamma_r^{(s)} \mapsto \Gamma_r^{(s)}$ such that $\Lambda^{(s+)}(w) = \frac{\partial v}{\partial v_r^{(s)}}$, where v is the solution of the system

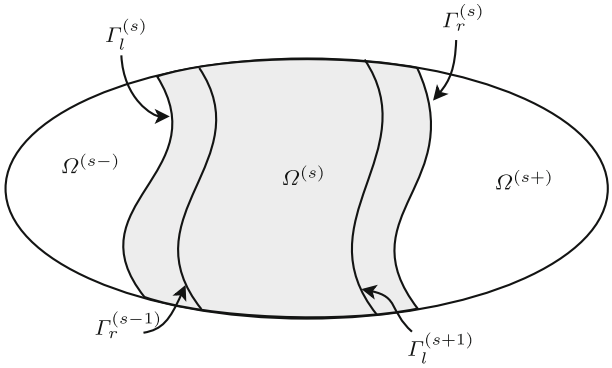

Fig. 1 Notation used in the one-way domain splitting

$$\begin{cases} \mathcal{L}(v) = 0 \quad \text{in } \Omega^{(s+)}, \\ v = w \quad \text{on } \Gamma_r^{(s)}, \\ \mathcal{C}(v) = 0 \quad \text{on } \partial\Omega \cap \partial\Omega^{(s+)}, \end{cases} \tag{8}$$

with $\nu_l^{(s)}$ (resp. $\nu_r^{(s)}$) denoting the unitary outward normal vector along $\Gamma_l^{(s)}$ (resp. $\Gamma_r^{(s)}$), i.e., pointing from $\Omega^{(s)}$ to $\Omega^{(s-1)}$ (resp. $\Omega^{(s+1)}$).

Remark 1 As mentioned earlier, there may be some overlap between the subdomains. For the convergence proofs in this paper, this overlap may be minimal, and consisting only of the artificial interfaces themselves, (this is often referred to as the "non-overlap" case), i.e., we may have $\Gamma_l^{(s)}$ and $\Gamma_r^{(s-1)}$ coinciding, and similarly, $\Gamma_r^{(s)}$ and $\Gamma_l^{(s+1)}$ coinciding as well.

We present now one synchronous iteration of the Schwarz algorithm with the interface conditions defined by the Steklov–Poincaré operators, for the solution of the problem (1). For each subdomain s, $s = 1, \ldots, p$, let $u^{(s)}(n+1)$ be the solution of

$$\begin{cases} \mathcal{L}\left(u^{(s)}(n+1)\right) = 0 \quad \text{in } \Omega^{(s)}, \\ \mathcal{C}(u^{(s)}(n+1)) = 0 \quad \text{on } \partial\Omega \cap \partial\Omega^{(s)}, \\ \left(\frac{\partial}{\partial \nu_l^{(s)}} - \Lambda^{(s-)}\right) u^{(s)}(n+1) = \left(\frac{\partial}{\partial \nu_l^{(s)}} - \Lambda^{(s-)}\right) u^{(s-1)}(n) \quad \text{on } \Gamma_l^{(s)}, \\ \left(\frac{\partial}{\partial \nu_r^{(s)}} - \Lambda^{(s+)}\right) u^{(s)}(n+1) = \left(\frac{\partial}{\partial \nu_r^{(s)}} - \Lambda^{(s+)}\right) u^{(s+1)}(n) \quad \text{on } \Gamma_r^{(s)}. \end{cases} \tag{9}$$

The Steklov–Poincaré operators are the Dirichlet to Neumann maps which give what is sometimes called *exact absorbing boundary* (interface) conditions [6]. It was shown in [32] that in the case of one-way splitting as described above, with overlap, the synchronous iterations of the Schwarz algorithm (9) converges in at most $p-1$ iterations, and this is why they are called optimal interface conditions (or optimal Schwarz methods). In order to demonstrate the convergence of the asynchronous Schwarz algorithm with optimal interface conditions, we first introduce the following lemma.

Lemma 1 *The two following statements hold.*

(**a**) *Let $u^{(s)}$ be the solution of the system*

$$\begin{cases} \mathcal{L}\left(u^{(s)}\right) = 0 \quad \textit{in } \Omega^{(s)}, \\ \mathcal{C}(u^{(s)}) = 0 \quad \textit{on } \partial\Omega \cap \partial\Omega^{(s)}, \\ \left(\frac{\partial}{\partial \nu_r^{(s)}} - \Lambda^{(s+)}\right) u^{(s)} = 0 \quad \textit{on } \Gamma_r^{(s)}. \end{cases} \tag{10}$$

Then the following relation is satisfied

$$\left(\frac{\partial}{\partial \nu_r^{(s-1)}} - \Lambda^{((s-1)+)}\right) u^{(s)} = 0 \quad \textit{on } \Gamma_r^{(s-1)}.$$

(**b**) *Let $u^{(s)}$ be the solution of the system*

$$\begin{cases} \mathcal{L}\left(u^{(s)}\right) = 0 \quad in\ \Omega^{(s)}, \\ \mathcal{C}(u^{(s)}) = 0 \quad on\ \partial\Omega \cap \partial\Omega^{(s)}, \\ \left(\frac{\partial}{\partial \nu_l^{(s)}} - \Lambda^{(s-)}\right) u^{(s)} = 0 \quad on\ \Gamma_l^{(s)}. \end{cases}$$

Then the following relation is satisfied

$$\left(\frac{\partial}{\partial \nu_l^{(s+1)}} - \Lambda^{((s+1)-)}\right) u^{(s)} = 0 \quad on\ \Gamma_l^{(s+1)}.$$

Proof Consider first case **(a)**. Suppose that $u^{(s)}$ satisfies Eq. (10). Let $v \in \Omega^{((s-1)+)}$ be the solution of

$$\begin{cases} \mathcal{L}(v) = 0 \quad \text{in } \Omega^{((s-1)+)}, \\ \mathcal{C}(v) = 0 \quad \text{on } \partial\Omega \cap \partial\Omega^{((s-1)+)}, \\ v = u^{(s)} \quad \text{on } \Gamma_r^{(s-1)}. \end{cases} \tag{11}$$

By the definition of $\Lambda^{((s-1)+)}$ given in (8), we have that

$$\frac{\partial v}{\partial \nu_r^{(s-1)}} = \Lambda^{((s-1)+)}(u^{(s)}) \quad \text{on } \Gamma_r^{(s-1)}.$$

Since $\Omega^{(s+)} \subset \Omega^{((s-1)+)}$, it holds that

$$\begin{cases} \mathcal{L}(v) = 0 \quad \text{in } \Omega^{(s+)}, \\ \mathcal{C}(v) = 0 \quad \text{on } \partial\Omega \cap \partial\Omega^{(s+)}. \end{cases}$$

Note that v is thus defined on $\Gamma_r^{(s)}$, so, it can be used as the known boundary condition in the system (8). As a consequence, with the definition of $\Lambda^{(s+)}$, we have that

$$\left(\frac{\partial}{\partial \nu_r^{(s)}} - \Lambda^{(s+)}\right) v = 0 \text{ on } \Gamma_r^{(s)}.$$

Thus, comparing (10) and (11), we have that $u^{(s)}$ and v are the solution of the same boundary value problem in $\Omega^{((s-1)+)} \cap \Omega^{(s)}$, in other words

$$\begin{cases} \mathcal{L}(v) = \mathcal{L}(u^{(s)}) = 0 \quad \text{in } \Omega^{((s-1)+)} \cap \Omega^{(s)}, \\ \mathcal{C}(v) = \mathcal{C}(u^{(s)}) = 0 \quad \text{on } \partial\Omega \cap \partial\left(\Omega^{((s-1)+)} \cap \Omega^{(s)}\right), \\ \left(\frac{\partial}{\partial \nu_r^{(s)}} - \Lambda^{(s+)}\right) v = \left(\frac{\partial}{\partial \nu_r^{(s)}} - \Lambda^{(s+)}\right) u^{(s)} = 0 \quad \text{on } \Gamma_r^{(s)}, \\ v = u^{(s)} \quad \text{on } \Gamma_r^{(s-1)}. \end{cases}$$

Since we assumed that the problem is well-posed, we have $u^{(s)} = v$ on $\Omega^{((s-1)+)} \cap \Omega^{(s)}$ and thus

$$\left(\frac{\partial}{\partial \nu_r^{(s-1)}} - \Lambda^{((s-1)+)} \right) u^{(s)} = 0 \quad \text{on } \Gamma_r^{(s-1)},$$

concluding the proof of the case (**a**) of the lemma.
Case (**b**) can be proved in a similar manner. □

Corollary 1 *For $s = p$, since $\Gamma_r^{(p)} = \emptyset$, the case* (**a**) *of Lemma 1 reduces to*

$$\begin{cases} \mathcal{L}\left(u^{(p)}\right) = 0 & \text{in } \Omega^{(p)} \\ \mathcal{C}(u^{(p)}) = 0 & \text{on } \partial\Omega \cap \partial\Omega^{(p)} \end{cases} \Rightarrow \left(\frac{\partial}{\partial \nu_r^{(p-1)}} - \Lambda^{((p-1)+)} \right) u^{(p)} = 0 \quad \text{on } \Gamma_r^{(p-1)}.$$

For $s = 1$, since $\Gamma_l^{(1)} = \emptyset$, the case (**b**) *of* Lemma 1 *reduces to*

$$\begin{cases} \mathcal{L}\left(u^{(1)}\right) = 0 & \text{in } \Omega^{(1)} \\ \mathcal{C}(u^{(1)}) = 0 & \text{on } \partial\Omega \cap \partial\Omega^{(1)} \end{cases} \Rightarrow \left(\frac{\partial}{\partial \nu_l^{(2)}} - \Lambda^{(2-)} \right) u^{(1)} = 0 \quad \text{on } \Gamma_l^{(2)}.$$

Proof This can be derived directly from the definition of the Steklov–Poincaré operators equation (8), since the boundary condition $v = u^{(p)}$ on $\Gamma_l^{(p)}$ (resp., $v = u^{(1)}$ on $\Gamma_r^{(1)}$) always holds. □

The (synchronous) algorithm defined in (9) can be rewritten as a fixed point iteration of the variables $z_l^{(s)}$ and $z_r^{(s)}$ for $s \in \{1, \ldots, p\}$, which represent the difference between the outward normal of the approximation $u^{(s)}$ and the image of the Steklov–Poincaré operator on $\Gamma_l^{(s)}$ and $\Gamma_r^{(s)}$, respectively; see further below. To that end, consider the map $\mathcal{T}^{(s)}$ defined on each subdomain s as

$$\mathcal{T}^{(s)} : (z_l^{(s+1)}(n), z_r^{(s-1)}(n)) \mapsto (z_l^{(s)}(n+1), z_r^{(s)}(n+1)),$$

where $(z_l^{(s+1)}(n), z_r^{(s-1)}(n))$ are the given boundary conditions for the problem

$$\begin{cases} \mathcal{L}\left(u^{(s)}(n+1)\right) = 0 & \text{in } \Omega^{(s)}, \\ \mathcal{C}(u^{(s)}(n+1)) = 0 & \text{on } \partial\Omega \cap \partial\Omega^{(s)}, \\ \left(\frac{\partial}{\partial \nu_l^{(s)}} - \Lambda^{(s-)} \right) u^{(s)}(n+1) = z_r^{(s-1)}(n) & \text{on } \Gamma_l^{(s)}, \\ \left(\frac{\partial}{\partial \nu_r^{(s)}} - \Lambda^{(s+)} \right) u^{(s)}(n+1) = z_l^{(s+1)}(n) & \text{on } \Gamma_r^{(s)}, \end{cases}$$

and we then compute

$$\begin{cases} z_l^{(s)}(n+1) = \left(\frac{\partial}{\partial \nu_r^{(s-1)}} - \Lambda^{((s-1)+)} \right) u^{(s)}(n+1), \\ z_r^{(s)}(n+1) = \left(\frac{\partial}{\partial \nu_l^{(s+1)}} - \Lambda^{((s+1)-)} \right) u^{(s)}(n+1). \end{cases} \tag{12}$$

The fixed point iteration starting with given values for $z_r^{(s)}(0)$ and $z_l^{(s)}(0)$ is then defined for each $s = 1, \ldots, p$, as

$$(z_l^{(s)}(n+1), z_r^{(s)}(n+1)) = \mathcal{T}^{(s)}\left((z_l^{(s+1)}(n), z_r^{(s-1)}(n))\right), \tag{13}$$

and it is easily seen that it is equivalent to the iteration (9). We shall see that this iteration converges to the fixed point of $\mathcal{T} = (\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(p)})$.

The iteration (13) is synchronous. Here we also consider an asynchronous version of it. In other words, we consider the operator (13) from the boundary conditions at a given time, to new boundary conditions (using local solves), with the optimal boundary conditions defined by the Steklov–Poincaré operator. In the next result, we prove that this asynchronous iteration is convergent. The proof consists of defining the appropriate spaces satisfying Assumption 2 and applying Theorem 3.

Theorem 5 *An asynchronous Schwarz algorithm defined by* (4) *with the operator* $\mathcal{T}$ *defined by* (13) *and the optimal interface condition* (13) *converges for any initial approximation.*

Proof We use Theorem 3. For each subdomain s, and for each time stamp $n = 1, 2, \ldots$, we define the sets

$$X_r^{(s)}(n) = \begin{cases} \mathbb{R}^{n_r^{(s)}} & \text{for } s > n \\ 0 & \text{else} \end{cases} \quad \text{and} \quad X_l^{(s)}(n) = \begin{cases} \mathbb{R}^{n_l^{(s)}} & \text{for } (p-s) \geq n \\ 0 & \text{else} \end{cases}$$

where $n_r^{(s)}$ (respectively $n_l^{(s)}$) is the number of degrees of freedom associated with $\Gamma_r^{(s)}$ (respectively $\Gamma_l^{(s)}$). We consider their products

$$X(n) = X_r^{(1)}(n) \times \prod_{s=2,\ldots,p-1} \left(X_l^{(s)}(n) \times X_r^{(s)}(n)\right) \times X_l^{(p)}(n).$$

To demonstrate the theorem, we will prove that $\mathcal{T}$ and X satisfy Assumption 2.

By definition, the sets $X(n)$, $X_l^{(s)}(n)$ and $X_r^{(s)}(n)$ satisfy the *Box Condition* of Assumption 2. Indeed, by contruction $X(n)$ is a Cartesian product space.

Now let us prove that the *Synchronous Convergence Condition* of Assumption 2 is also satisfied. For every n, we have (by construction)

$$X_r^{(s)}(n+1) \subset X_r^{(s)}(n) \quad \text{and} \quad X_l^{(s)}(n+1) \subset X_l^{(s)}(n)$$

which leads to

$$X(n+1) \subset X(n).$$

Let us choose an arbitrary n and define

$$z(n) = (z_r^{(1)}(n), \ldots, z_l^{(s)}(n), z_r^{(s)}(n), \ldots, z_l^{(p)}(n)) \in X(n)$$

and

$$z(n+1) = \mathcal{T}(z(n)) = (z_r^{(1)}(n+1), \ldots, z_l^{(s)}(n+1), z_r^{(s)}(n+1), \ldots, z_l^{(p)}(n+1)).$$

First we show that $\forall s, z_r^{(s)}(n+1) \in X_r^{(s)}(n+1)$. We have:

- For $1 < s \leq p$,
  - if $s > n+1$ then $X_r^{(s)}(n+1) = \mathbb{R}^{n_r^{(s)}}$ so $z_r^{(s)}(n+1) \in X_r^{(s)}(n+1)$.
  - if $s \leq n+1$ then $s-1 \leq n$ and $X_r^{(s-1)}(n) = 0$, i.e., $z_r^{(s-1)}(n) = 0$. Lemma 1 implies that $\mathcal{T}^{(s)}(\cdot, 0) = (\cdot, 0)$ so $z_r^{(s)}(n+1) = 0$.
- For $s = 1$, Corollary 1 implies that $z_r^{(1)}(n+1) = 0$ without condition, so $z_r^{(1)}(n+1) \in X_r^{(1)}(n+1)$.

The same proof can be used to demonstrate that

$$\forall s, z_l^{(s)}(n+1) \in X_l^{(s)}(n+1).$$

This leads to $z(n+1) = \mathcal{T}(z(n)) \in X(n+1)$. Moreover, $\forall n > p$, $X(n) = 0$ and 0 is a fixed point of $\mathcal{T}$. So the *Synchronous Convergence Condition* of Assumption 2 is satisfied.
Thus, we can apply now Theorem 3, which guaranties the convergence of the algorithm for any initial approximation, and this concludes the proof. □

The Schwarz algorithm with optimal interface conditions corresponds to an iterative procedure which is formulated on the interface. The demonstration of the convergence of the iterates on the interface conditions is the key point for the convergence of the iterates within the subdomains. As already mentioned in [6], optimal interface conditions lead to non-local differential operators, making the implementation difficult. Therefore, for practical implementation of these methods one needs to consider approximations to these optimal conditions, e.g., optimized interface conditions. These optimized conditions are local, that is, they depend only on neighboring subdomains. We do so in the next section.

4 Asynchronous Schwarz with optimized interface conditions

In order to prove the convergence of Schwarz algorithm with optimized interface conditions, we consider a simpler theoretical case. The aim is to solve the general equation

$$\Delta u = f \text{ on } \mathbb{R}^2 \tag{14}$$

with vanishing value of u at infinity. The space $\mathbb{R}^2$ is divided into p overlapping infinite vertical strips. This means we have $p-1$ vertical lines, say at coordinates $x = \ell_1, \ldots \ell_{p-1}$; and we assume for simplicity that we have the same overlap $2L$ between the subdomains. We also assume, without loss of generality, that except for the subdomains at infinity, that each strip has the same width, i.e., $\ell_s - \ell_{s-1} = W$ for $s = 2, \ldots, p-1$, so that $\ell_s = \ell_1 + (s-1)W$. It follows then, that the overlap satisfies

$2L < W$. Thus, we have $\Omega^{(1)} =]-\infty; \ell_1 + L] \times \mathbb{R}$, $\Omega^{(s)} = [\ell_{s-1} - L; \ell_s + L] \times \mathbb{R}$, $s = 2, \ldots, p-1$, and $\Omega^{(p)} = [\ell_{p-1} - L; +\infty[\times \mathbb{R}$. In this context, the normal vector is in the x direction (with the appropriate sign). In order to analyze convergence of an asynchronous optimized Schwarz approach for the Laplacian problem (14), we first prove convergence for a nearby problem, namely the shifted Laplacian

$$\Delta u - \eta = f \text{ on } \mathbb{R}^2, \tag{15}$$

with vanishing value of u at infinity, and $\eta > 0$.

Let $f^{(s)}$ and $u^{(s)}(n)$ denote the restriction of f and $u(n)$, the approximation to the solution at the iteration n, to $\Omega^{(s)}$, $s = 1, \ldots, p$, respectively. Thus, $u^{(s)}(n) \in V^{(s)}$, a space of functions defined on $\Omega^{(s)}$. We consider the transmission operator Λ defined on the interfaces, i.e., on the vertical lines.

The operator Λ here is rather general, and, e.g., it can represent the OO0 condition by setting $\Lambda = \alpha u$, or the OO2 condition by setting $\Lambda = \alpha u + \beta \frac{\partial^2 u}{\partial \tau^2}$, for $\alpha, \beta \in \mathbb{R}$.

Our approach to demonstrate the convergence of the asynchronous iterations in the case of optimized conditions, i.e., of the optimized version of the iteration (9), is different than that in the previous section. We show that the iteration operator is convergent and apply Theorem 4. To that end, a Fourier transform is applied along the y axis, as initially suggested in [6] for the Maxwell equation (in the synchronous case). The spatial frequency variable is designed by k, with $k \in \mathbb{R}$. In other words, we have

$$\hat{u}^{(s)}(x, k) = \int_{-\infty}^{\infty} e^{-iky} u^{(s)}(x, y) dy.$$

Thus, derivatives of u with respect to y become multiplications by $-ik$ in the Fourier space. The goal is then to show that the Fourier transform of the asynchronous iteration converges to zero. We show that the component for each of the frequencies goes to zero with increasing n. The operator $\lambda(k)$ represents the Fourier transform of the operator Λ. Thus, in the OO0 case, $\lambda(k) = \alpha$, while in the OO2 case, $\lambda(k) = \alpha + \beta k^2$. We begin by showing that the synchronous Schwarz has a contracting operation in the Fourier space from one iteration to the next.

We denote by $\theta(k)$ the positive root of the characteristic equation of (15), namely of the equation $\theta^2 - (\eta + k^2) = 0$. That is, $\theta(k) = \sqrt{\eta + k^2}$, which is well defined for $\eta > 0$.

Theorem 6 *For the shifted Laplacian problem (15), and a domain decomposition with p vertical strips of width W for $p = 2, \ldots, p-1$, overlap $2L$ throughout, and transmition operator Λ defined on the interfaces, supposed to be the same on the left and right interfaces, given an initial approximation $u(0) = (u^{(1)}(0), \ldots, u^{(p)}(0)) \in V^{(1)} \times V^{(2)} \times \cdots \times V^{(p)}$, we consider the following synchronous optimized Schwarz algorithm*

$$
\begin{cases}
(\Delta-\eta)u^{(1)}(n+1)=f^{(1)} & \text{on } \Omega^{(1)},\\
\frac{\partial u^{(1)}(n+1)}{\partial x}+\Lambda(u^{(1)}(n+1))=\frac{\partial u^{(2)}(n)}{\partial x}+\Lambda(u^{(2)}(n)) & \text{for } x=\ell_1+L,\\
\text{For } s=2,\ldots,p-1, & \\
-\frac{\partial u^{(s)}(n+1)}{\partial x}+\Lambda(u^{(s)}(n+1))=-\frac{\partial u^{(s-1)}(n)}{\partial x}+\Lambda(u^{(s-1)}(n)) & \text{for } x=\ell_{s-1}-L,\\
(\Delta-\eta)u^{(s)}(n+1)=f^{(s)} & \text{on } \Omega^{(s)},\\
\frac{\partial u^{(s)}(n+1)}{\partial x}+\Lambda(u^{(s)}(n+1))=\frac{\partial u^{(s+1)}(n)}{\partial x}+\Lambda(u^{(s+1)}(n)) & \text{for } x=\ell_s+L,\\
-\frac{\partial u^{(p)}(n+1)}{\partial x}+\Lambda(u^{(p)}(n+1))=-\frac{\partial u^{(p-1)}(n)}{\partial x}+\Lambda(u^{(p)}(n)) & \text{for } x=\ell_{p-1}-L,\\
(\Delta-\eta)u^{(p)}(n+1)=f^{(p)} & \text{on } \Omega^{(p)},
\end{cases}
\tag{16}
$$

We represent this iteration by the operators $\mathcal{T}^{(s)}: V^{(1)}\times\cdots\times V^{(p)}\to V^{(s)}$, *so that* $\mathcal{T}^{(s)}(u^{(1)}(n),\ldots,u^{(p)}(n))=u^{(s)}(n+1)$. *Let* $e^{-\sqrt{\eta}W}<\phi<1$ *be fixed. Let* $\lambda(k)$ *be the Fourier transform of the (Robin type) operator* Λ, *for the frequency* k, *and assume that it satisfies the following two conditions*

$$
\text{for all } k,\quad \lambda(k)\neq-\theta(k),\quad \text{and} \tag{17}
$$

$$
\frac{|\lambda(k)-\theta(k)|}{|\lambda(k)+\theta(k)|}\cdot e^{-\theta(k)2L}<\frac{\phi-e^{-\theta(k)W}}{1+\phi e^{-\theta(k)W}}\cdot \tag{18}
$$

Then, the algorithm converges to the solution u *for any initial approximation* $u(0)$.

Proof For this proof, without loss of generality, we consider the homogeneous case ($f=0$). Since the differential operator is linear, this means that we are considering the error, i.e., the difference between the solution and the approximation $u(n)$, and we will show that this error, denoted again by $u(n)$ goes to zero. We apply a Fourier transform along the y axis. Therefore for Eq. (15) on a generic subdomain $\Omega^{(s)}$,

$$
\frac{\partial^2\hat{u}^{(s)}}{\partial x^2}-\eta\hat{u}^{(s)}+i^2k^2\hat{u}^{(s)}=\frac{\partial^2\hat{u}^{(s)}}{\partial x^2}-(\eta+k^2)\hat{u}^{(s)}=0. \tag{19}
$$

Before we give the proof in full detail, we present a sketch of it. 1. Using the Fourier transform (19), we write the Fourier version of the algorithm (16), for each frequency k. 2. We then write this Fourier transform of the approximate solution in each subdomain at each iteration n, in terms of two exponentials, with the coefficients in front of these two exponentials, say $A_s^n(k)$ and $B_s^n(k)$ to be determined. 3. We use the Robin-type boundary conditions on the interfaces to write relations between these coefficients at iteration $n+1$, as a function of those at iteration n. There are dependencies from the corresponding coefficient in the subdomains on the left and right of the subdomain. 4. This results in a linear map from the coefficients of the Fourier transform for all subdomains $s=1,\ldots,p$, at iteration n to all coefficients at iteration $n+1$. We explicitly write the matrix $\hat{\mathcal{T}}$ of this linear map, and consider its max-norm. 5. We show that this norm is less than $\phi<1$, and thus, the map is a contraction impying that $\hat{u}(n)\to 0$. 6. We relate the operator $\mathcal{T}$ to the matrix $\hat{\mathcal{T}}$, showing that the former is convergent and thus $u(n)\to 0$, completing the proof.

Step 1. Using (19), we obtain from (16), for each frequency k

$$\begin{cases} \frac{\partial^2 \hat{u}^{(1)}(n+1)}{\partial x^2} - (\eta + k^2)\hat{u}^{(1)}(n+1) = 0 & \text{for } x \le \ell_1 + L, \\ \frac{\partial \hat{u}^{(1)}(n+1)}{\partial x} + \lambda \hat{u}^{(1)}(n+1) = \frac{\partial \hat{u}^{(2)}(n)}{\partial x} + \lambda \hat{u}^{(2)}(n) & \text{for } x = \ell_1 + L, \\ \text{For } s = 2, \ldots, p-1, & \\ -\frac{\partial \hat{u}^{(s)}(n+1)}{\partial x} + \lambda \hat{u}^{(s)}(n+1) = -\frac{\partial \hat{u}^{(s-1)}(n)}{\partial x} + \lambda \hat{u}^{(s-1)}(n) & \text{for } x = \ell_{s-1} - L, \\ \frac{\partial^2 \hat{u}^{(s)}(n+1)}{\partial x^2} - (\eta + k^2)\hat{u}^{(s)}(n+1) = 0 & \text{for } \ell_{s-1} - L \le x \le \ell_s + L, \\ \frac{\partial \hat{u}^{(s)}(n+1)}{\partial x} + \lambda \hat{u}^{(s)}(n+1) = \frac{\partial \hat{u}^{(s+1)}(n)}{\partial x} + \lambda \hat{u}^{(s+1)}(n) & \text{for } x = \ell_s + L, \\ -\frac{\partial \hat{u}^{(p)}(n+1)}{\partial x} + \lambda \hat{u}^{(p)}(n+1) = -\frac{\partial \hat{u}^{(p-1)}(n)}{\partial x} + \lambda \hat{u}^{(p-1)}(n) & \text{for } x = \ell_{p-1} - L, \\ \frac{\partial^2 \hat{u}^{(p)}(n+1)}{\partial x^2} - (\eta + k^2)\hat{u}^{(p)}(n+1) = 0 & \text{for } \ell_{p-1} - L \le x, \end{cases} \tag{20}$$

where $\hat{u}$ denotes the Fourier transform of u. The quantity λ depends on k, and $\hat{u}^{(s)}(n) = \hat{u}^{(s)}(n)(x,k)$ depends on x and k; however, in the interest of clarity, one or both of the parameters x and k are sometimes omitted in our presentation.

Step 2. A general solution of (20) has the form

$$\hat{u}^{(s)}(n)(x,k) = A_s^n(k)e^{\theta_1(x-(\ell_{s-1}-L))} + B_s^n(k)e^{\theta_2(x-(\ell_s+L))}, \quad s = 1, \ldots, p, \tag{21}$$

where θ_i are the two solutions of the characteristic equation of the operator, namely, $\theta_1 = -\theta(k), \theta_2 = \theta(k)$. The vanishing condition of the solution at infinity, thus imply that $A_1^n(k) = B_p^n(k) = 0$.

Therefore, the solutions of (20) have the following form

$$\begin{cases} \hat{u}^{(1)}(n)(x,k) = B_1^n(k)e^{\theta(k)(x-(\ell_1+L))}, \\ \hat{u}^{(s)}(n)(x,k) = A_s^n(k)e^{-\theta(k)(x-(\ell_{s-1}-L))} + B_s^n(k)e^{\theta(k)(x-(\ell_s+L))}, \quad s = 2, \ldots, p-1, \\ \hat{u}^{(p)}(n)(x,k) = A_p^n(k)e^{-\theta(k)(x-(\ell_{p-1}-L))}, \end{cases}$$

with the unknown coefficients $B_1^n(k)$, $A_s^n(k)$, $B_s^n(k)$, $A_p^n(k)$ to be determined. From now on, we drop the parameter k in the expression of these coefficients. Direct computation of the derivatives and substitutions lead to the following equations

$$\begin{cases} \frac{\partial \hat{u}^{(1)}(n)(x,k)}{\partial x} = \theta(k)\hat{u}^{(1)}(n)(x,k), \\ \frac{\partial \hat{u}^{(s)}(n)(x,k)}{\partial x} = -\theta(k)A_s^n e^{-\theta(k)(x-(\ell_{s-1}-L))} + \theta(k)B_s^n e^{\theta(k)(x-(\ell_s+L))}, \quad s = 2, \ldots, p-1, \\ \frac{\partial \hat{u}^{(p)}(n)(x,k)}{\partial x} = -\theta(k)\hat{u}^{(p)}(n)(x,k). \end{cases}$$

Let us define

$$\kappa = (\lambda - \theta(k)) \quad \text{and} \quad \mu = (\lambda + \theta(k)).$$

Since $\lambda \neq -\theta(k)$, this leads to $\mu \neq 0$.

Step 3. The introduction of these expressions in the interface conditions gives us, for subdomain $\Omega^{(1)}$ receiving values from subdomain $\Omega^{(2)}$,

$$\begin{aligned}
&\frac{\partial \hat{u}^{(1)}(n+1,\ell_1+L)}{\partial x}+\lambda\hat{u}^{(1)}(n+1,\ell_1+L)=\frac{\partial \hat{u}^{(2)}(n,\ell_1+L)}{\partial x}+\lambda\hat{u}^{(2)}(n,\ell_1+L),\\
&(\theta(k)+\lambda)\,B_1^{n+1}=-\theta(k)A_2^n e^{-\theta(k)2L}+\theta(k)B_2^n e^{-\theta(k)W}+\lambda A_2^n e^{-\theta(k)2L}\\
&\quad+\lambda B_2^n e^{-\theta(k)(\ell_2-\ell_1)},\\
&B_1^{n+1}=\frac{\kappa}{\mu}A_2^n e^{-\theta(k)2L}+B_2^n e^{-\theta(k)W};
\end{aligned}$$

for subdomain $\Omega^{(p)}$ receiving values from subdomain $\Omega^{(p-1)}$,

$$\begin{aligned}
&-\frac{\partial \hat{u}^{(p)}(n+1,\ell_{p-1}-L)}{\partial x}+\lambda\hat{u}^{(p)}(n+1,\ell_{p-1}-L)\\
&\quad=-\frac{\partial \hat{u}^{(p-1)}(n,\ell_{p-1}-L)}{\partial x}+\lambda\hat{u}^{(p-1)}(n,\ell_{p-1}-L),\\
&(\theta(k)+\lambda)\,A_p^{n+1}=\theta(k)A_{p-1}^n e^{-\theta(k)(\ell_{p-1}-\ell_{p-2})}-\theta(k)B_{p-1}^n e^{-\theta(k)2L}\\
&\quad+\lambda A_{p-1}^n e^{-\theta(k)(\ell_{p-1}-\ell_{p-2})}+\lambda B_{p-1}^n e^{-\theta(k)2L},\\
&A_p^{n+1}=A_{p-1}^n e^{-\theta(k)W}+\frac{\kappa}{\mu}B_{p-1}^n e^{-\theta(k)2L};
\end{aligned}$$

for subdomain $\Omega^{(2)}$ receiving values from subdomain $\Omega^{(1)}$,

$$\begin{aligned}
&-\frac{\partial \hat{u}^{(2)}(n+1)(\ell_1-L)}{\partial x}+\lambda\hat{u}^{(2)}(n+1)(\ell_1-L)\\
&\quad=-\frac{\partial \hat{u}^{(1)}(n)(\ell_1-L)}{\partial x}+\lambda\hat{u}^{(1)}(n)(\ell_1-L),\\
&\theta(k)A_2^{n+1}-\theta(k)B_2^{n+1}e^{-\theta(k)(\ell_2-\ell_1+2L)}+\lambda A_2^{n+1}+\lambda B_2^{n+1}e^{-\theta(k)(\ell_2-\ell_1+2L)}\\
&\quad=(-\theta(k)+\lambda)\,B_1^n e^{-\theta(k)2L},\\
&A_2^{n+1}+\frac{\kappa}{\mu}e^{-\theta(k)(W+2L)}B_2^{n+1}=\frac{\kappa}{\mu}B_1^n e^{-\theta(k)2L};
\end{aligned}\tag{22}$$

for subdomain $\Omega^{(p-1)}$ receiving values from subdomain $\Omega^{(p)}$,

$$\begin{aligned}
&\frac{\partial \hat{u}^{(p-1)}(n+1,\ell_{p-1}+L)}{\partial x}+\lambda\hat{u}^{(p-1)}(n+1,\ell_{p-1}+L)\\
&\quad=\frac{\partial \hat{u}^{(p)}(n,\ell_{p-1}+L)}{\partial x}+\lambda\hat{u}^{(p)}(n,\ell_{p-1}+L),\\
&\quad-\theta(k)A_{p-1}^{n+1}e^{-\theta(k)(\ell_{p-1}-\ell_{p-2}+2L)}+\theta(k)B_{p-1}^{n+1}+\lambda A_{p-1}^{n+1}e^{-\theta(k)(\ell_{p-1}-\ell_{p-2}+2L)}+\lambda B_{p-1}^{n+1}\\
&\quad=(-\theta(k)+\lambda)\,A_p^n e^{-\theta(k)2L},\\
&\frac{\kappa}{\mu}e^{-\theta(k)(W+2L)}A_{p-1}^{n+1}+B_{p-1}^{n+1}=\frac{\eta}{\mu}A_p^n e^{-\theta(k)2L};
\end{aligned}\tag{23}$$

for $s = 3, \ldots, p-1$, we have for subdomain $\Omega^{(s)}$ receiving values from subdomain $\Omega^{(s-1)}$,

$$
\begin{aligned}
&-\frac{\partial \hat{u}^{(s)}(n+1, \ell_{s-1}-L)}{\partial x} + \lambda \hat{u}^{(s)}(n+1, \ell_{s-1}-L) \\
&\quad = -\frac{\partial \hat{u}^{(s-1)}(n)(n, \ell_{s-1}-L)}{\partial x} + \lambda \hat{u}^{(s-1)}(n)(n, \ell_{s-1}-L), \\
&(\theta(k)+\lambda)\, A_s^{n+1} + (\lambda - k) B_s^{n+1} e^{-\theta(k)(\ell_s - \ell_{s-1}+2L)} \\
&\quad = \theta(k) A_{s-1}^n e^{-\theta(k)(\ell_{s-1}-\ell_{s-2})} - \theta(k) B_{s-1}^n e^{-\theta(k)2L} \\
&\qquad + \lambda A_{s-1}^n e^{-\theta(k)(\ell_{s-1}-\ell_{s-2})} + \lambda B_{s-1}^n e^{-\theta(k)2L}, \\
&A_s^{n+1} + \frac{\kappa}{\mu} e^{-\theta(k)(W+2L)} B_s^{n+1} = A_{s-1}^n e^{-\theta(k)W} + \frac{\kappa}{\mu} B_{s-1}^n e^{-\theta(k)2L}; \qquad (24)
\end{aligned}
$$

and for $s = 2, \ldots, p-2$, we have for subdomain $\Omega^{(s)}$ receiving values from subdomain $\Omega^{(s+1)}$,

$$
\begin{aligned}
&\frac{\partial \hat{u}^{(s)}(n+1, \ell_s+\delta)}{\partial x} + \lambda \hat{u}^{(s)}(n+1, \ell_s+L) \\
&\quad = \frac{\partial \hat{u}^{(s+1)}(n)(n, \ell_s+L)}{\partial x} + \lambda \hat{u}^{(s+1)}(n)(n, \ell_s+L), \\
&-\theta(k) A_s^{n+1} e^{-\theta(k)(\ell_s - \ell_{s-1}+2L)} + \theta(k) B_s^{n+1} + \lambda A_s^{n+1} e^{-\theta(k)(\ell_s - \ell_{s-1}+2L)} + \lambda B_s^{n+1} \\
&\quad = (-\theta(k)+\lambda)\, A_{s+1}^n e^{-\theta(k)2L} + (\theta(k)+\lambda) B_{s+1}^n e^{-\theta(k)(\ell_{s+1}-\ell_s)}, \\
&\frac{\kappa}{\mu} e^{-\theta(k)(W+2L)} A_s^{n+1} + B_s^{n+1} = \frac{\kappa}{\mu} A_{s+1}^n e^{-\theta(k)2L} + B_{s+1}^n e^{-\theta(k)W}. \qquad (25)
\end{aligned}
$$

Step 4. We now rewrite the 2×2 system formed by the two equations (24) and (25), which corresponds to the exchange between the two interfaces of the subdomain $\Omega^{(s)}$. Let $\gamma = \kappa/\mu = |\lambda - \theta(k)| \,/\, |\lambda + \theta(k)|$, we then have

$$
\begin{aligned}
&\begin{bmatrix} 1 & \gamma e^{-\theta(k)(W+2L)} \\ \gamma e^{-\theta(k)(W+2L)} & 1 \end{bmatrix} \begin{bmatrix} A_s^{n+1} \\ B_s^{n+1} \end{bmatrix} \\
&\quad = \begin{bmatrix} e^{-\theta(k)W} & \gamma e^{-\theta(k)2L} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} A_{s-1}^n \\ B_{s-1}^n \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \gamma e^{-\theta(k)2L} & e^{-\theta(k)W} \end{bmatrix} \begin{bmatrix} A_{s+1}^n \\ B_{s+1}^n \end{bmatrix}. \qquad (26)
\end{aligned}
$$

Let us denote by S the matrix on the left hand side, and by R_1 and R_2, the two matrices on the right hand side. Let us collect the $2p-2$ coefficients at the step n in the vector $c(n)^T = ((c_1(n), c_2(n), \ldots, c_{p-1}(n), c_p(n)) = (B_1^n, A_2^n, B_2^n, \ldots, A_{p-1}^n, B_{p-1}^n, A_p^n)$, where $c_1 = B_1^n$ and $c_p = A_p^n$ are scalars, and $c_s = (A_s^n, B_s^n)$ are ordered pairs for $s = 2, \ldots, p-1$. With this notation, we rewrite (26) as

$$
S c_s(n+1) = R_1 c_{s-1}(n) + R_2 c_{s+1}(n),
$$

$$
\begin{bmatrix}
1 & & & & & \\
& [S] & & & & \\
& & \ddots & & & \\
& & & \ddots & & \\
& & & & [S] & \\
& & & & & 1
\end{bmatrix}
\begin{bmatrix}
B_1^{n+1} \\
A_2^{n+1} \\
B_2^{n+1} \\
\vdots \\
\vdots \\
A_{p-1}^{n+1} \\
B_{p-1}^{n+1} \\
A_p^{n+1}
\end{bmatrix}
=
$$

$$
\begin{bmatrix}
0 & \begin{bmatrix} r_{21}^{(2)} & r_{22}^{(2)} \end{bmatrix} & & & & & & \\
\begin{bmatrix} R_1^{(2)} \end{bmatrix} & \begin{bmatrix} 0 \end{bmatrix} & \begin{bmatrix} R_2 \end{bmatrix} & & & & & \\
& \begin{bmatrix} R_1 \end{bmatrix} & \begin{bmatrix} 0 \end{bmatrix} & \begin{bmatrix} R_2 \end{bmatrix} & & & & \\
& & \ddots & \ddots & \ddots & & & \\
& & & \ddots & \ddots & \begin{bmatrix} R_2 \end{bmatrix} & & \\
& & & & \begin{bmatrix} R_1 \end{bmatrix} & \begin{bmatrix} 0 \end{bmatrix} & \begin{bmatrix} R_2^{(1)} \end{bmatrix} & \\
& & & & & \begin{bmatrix} r_{11}^{(1)} & r_{12}^{(1)} \end{bmatrix} & 0
\end{bmatrix}
\begin{bmatrix}
B_1^n \\
A_2^n \\
B_2^n \\
A_3^n \\
B_3^n \\
\vdots \\
A_{p-1}^n \\
B_{p-1}^n \\
A_p^n
\end{bmatrix}
$$

Fig. 2 Schematic representation of (27)

where S is symmetric positive definite, and thus nonsingular. Collecting these 2×2 systems into an $(2p-2) \times (2p-2)$ system, we write it as

$$\hat{\mathcal{S}} c(n+1) = \hat{\mathcal{R}} c(n), \tag{27}$$

which is shown schematically in Fig. 2, and where $\hat{\mathcal{S}}$ is a symmetric positive definite block diagonal matrix with first and last diagonal blocks being 1×1 with entry 1, and $(p-2)$ 2×2 diagonal blocks being the matrix S, and $\hat{\mathcal{R}}$ is a block tridiagonal matrix with blocks commensurate with those of $\hat{\mathcal{S}}$, and the typical block row is $[R_1 \mid O \mid R_2]$, with R_1 and R_2 appropriately truncated near the corners of the matrix, that is, $R_1^{(2)}$ and $R_2^{(1)}$ are the second column of R_1 and the first column of R_2, respectively, and $\begin{bmatrix} r_{21}^{(2)} & r_{22}^{(2)} \end{bmatrix}$ is the second row of R_2, and similarly, the first row of R_1 appearing in the last row of $\hat{\mathcal{R}}$. We use this notation to emphasize that these operators, as well as $\hat{\mathcal{T}}$

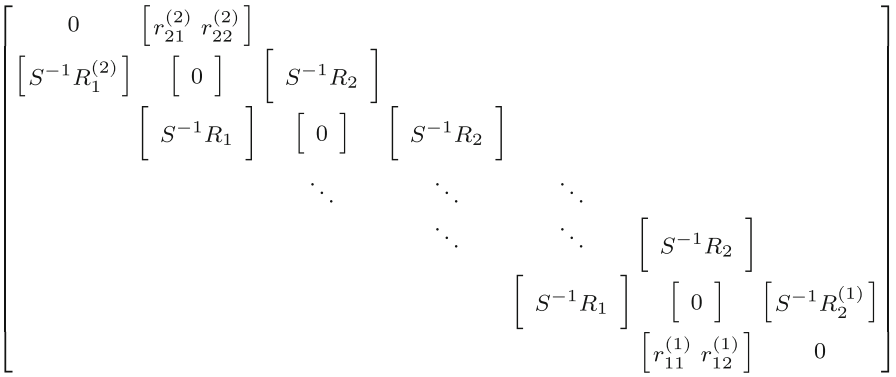

Fig. 3 Schematic representation of $\hat{\mathcal{T}}$

defined below act on coefficients in the Fourier space. In order to analyze the iteration (27), we rewrite it as

$$c(n+1) = \hat{\mathcal{T}} c(n),$$

where $\hat{\mathcal{T}} = \hat{\mathcal{S}}^{-1}\hat{\mathcal{R}}$, and it is schematically represented in Fig. 3. Since $\hat{\mathcal{S}}$ is block diagonal, so is its inverse. The first and last blocks of $\hat{\mathcal{S}}$ are 1×1 with the entry having value 1, and thus, this is also the case for $\hat{\mathcal{S}}^{-1}$. All other diagonal blocks being S^{-1}.

One can see by inspection that

$$S^{-1} = \frac{1}{1-\gamma^2 e^{-2\theta(k)(W+2L)}} \begin{bmatrix} 1 & -\gamma e^{-\theta(k)(W+2L)} \\ -\gamma e^{-\theta(k)(W+2L)} & 1 \end{bmatrix}.$$

We can then compute

$$S^{-1}R_1 = \frac{1}{1-\gamma^2 e^{-2\theta(k)(W+2L)}} \begin{bmatrix} e^{-\theta(k)W} & \gamma e^{-\theta(k)2L} \\ -\gamma e^{-\theta(k)W} e^{-\theta(k)(W+2L)} & -\gamma^2 e^{-\theta(k)2L} e^{-\theta(k)(W+2L)} \end{bmatrix},$$

$$S^{-1}R_2 = \frac{1}{1-\gamma^2 e^{-2\theta(k)(W+2L)}} \begin{bmatrix} -\gamma^2 e^{-\theta(k)2L} e^{-\theta(k)(W+2L)} & -\gamma e^{-\theta(k)W} e^{-\theta(k)(W+2L)} \\ \gamma e^{-\theta(k)2L} & e^{-\theta(k)W} \end{bmatrix}.$$

Step 5. Our goal is to show that $\hat{\mathcal{T}} = \hat{\mathcal{T}}(k)$ is contracting. To that end, we shall compute $\|\hat{\mathcal{T}}\|_\infty$, i.e., its maximum row sum, with elements taken in absolute value. A typical block row of the block tridiagonal matrix $\hat{\mathcal{T}}$ is of the form $[S^{-1}R_1 \mid O \mid S^{-1}R_2]$. Each of the two row sums of this typical block row is thus

$$\begin{aligned} &\frac{1}{1-\gamma^2 e^{-2\theta(k)(W+2L)}} \\ &\quad \times \left[e^{-\theta(k)W} + |\gamma| e^{-\theta(k)2L} + \gamma^2 e^{-\theta(k)2L} e^{-\theta(k)(W+2L)} + |\gamma| e^{-\theta(k)W} e^{-\theta(k)(W+2L)} \right] \\ &= \frac{1}{1-\gamma^2 e^{-2\theta(k)(W+2L)}} \end{aligned}$$

$$\times \left[e^{-\theta(k)W}(1+|\gamma|e^{-\theta(k)(W+2L)}) + |\gamma|e^{-\theta(k)2L}(1+|\gamma|e^{-\theta(k)(W+2L)}) \right]$$
$$= \frac{e^{-\theta(k)W} + |\gamma|e^{-\theta(k)2L}}{1-|\gamma|e^{-\theta(k)(W+2L)}}. \tag{28}$$

Since the first few and the last few rows have fewer terms of the same form, or have a smaller factor in front, the quantity (28) is indeed the maximum, i.e., $\|\hat{\mathcal{T}}(k)\|_\infty$.

The following are equivalent:

$$\begin{aligned}
\frac{e^{-\theta(k)W} + |\gamma|e^{-\theta(k)2L}}{1-|\gamma|e^{-\theta(k)(W+2L)}} &< \phi \\
e^{-\theta(k)W} + |\gamma|e^{-\theta(k)2L} &< \phi(1-|\gamma|e^{-\theta(k)(W+2L)}) \\
&= \phi - \phi|\gamma|e^{-\theta(k)2L}e^{-\theta(k)W} \\
e^{-\theta(k)W} + |\gamma|e^{-\theta(k)2L}(1+\phi e^{-\theta(k)W}) &< \phi \\
|\gamma|e^{-\theta(k)2L} &< \frac{\phi - e^{-\theta(k)W}}{1+\phi e^{-\theta(k)W}},
\end{aligned}$$

which is precisely our hypothesis (18).

Therefore, for each k, $\|\hat{\mathcal{T}}(k)\|_\infty < \phi < 1$.

Step 6. Let $\mathcal{I}_u$ be the isomorphism mapping the vector $c(n)$ to the tuple of functions in Fourier space $(\hat{u}^{(1)}, \ldots, \hat{u}^{(p)})$, and let $\mathcal{I}_c$ be its inverse. Denote by $\mathcal{F}$ the map from the tuple of functions $(u^{(1)}, \ldots, u^{(p)})$ to $(\hat{u}^{(1)}, \ldots, \hat{u}^{(p)})$. It follows then that

$$\mathcal{T} = (\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(p)}) = \mathcal{F}^{-1}\,\mathcal{I}_c\,\hat{\mathcal{T}}\,\mathcal{I}_u\,\mathcal{F},$$

and therefore

$$\mathcal{T}^n = \mathcal{F}^{-1}\,\mathcal{I}_c\,\hat{\mathcal{T}}^n\,\mathcal{I}_u\,\mathcal{F}.$$

The fact that $\hat{\mathcal{T}}$ is contracting, thus implies that $\mathcal{T}^n \to 0$. This concludes the proof. □

Remark 2 We discuss here the hypotheses (17)–(18). In all our experiments, the approximate optimal conditions give $\lambda = \alpha$ for OO0, or $\lambda = \alpha + \beta k^2$, with $\alpha > 0$ and $\beta > 0$, and thus the hypothesis (17) always hold. Regarding the second hypthesis (18), the first factor on the left is less than one. The second factor is less than one when the overlap $L > 0$, and equal to one when $L = 0$. The term on the right of the condition is also less than one, and it is a monotone increasing function of the width of the subdomains W. The smallest value it can take is when $k = 0$ and this value is

$$\frac{\phi - e^{-\sqrt{\eta}W}}{1+\phi e^{\sqrt{\eta}W}}$$

Thus, for a fixed amount of overlap $2L$ (which could be zero), for large enough W, i.e., for wide enough subdomains condition (18) is automatically satisfied.

Remark 3 While Theorem 6 was proved for an equation on the whole space $\mathbb{R}^2$, a similar proof would apply to an infinite cylinder, as was, e.g., in [9]. We also note that other than the mentioned result in that reference, this is the only proof we know of (synchronous) optimized Schwarz convergence for $p > 2$ subdomains for the shifted Laplacian.

Using the fact that Theorem 6 provides a (weighted) max-norm for which the error is contracting, we can now apply Theorem 4 to the asynchronous optimized Schwarz method for the shifted Laplacian, and thus we have a proof of its convergence. For completeness, we state the result as the following theorem.

Theorem 7 *For the shifted Laplacian problem* (15)*, and a domain decomposition with p vertical strips of width W for $s = 2, \ldots, p-1$, overlap $2L$ throughout, and transmition operator Λ defined on the interfaces, supposed to be the same on the left and rigth interfaces, given an initial approximation $u(0) = (u^{(1)}(0), \ldots, u^{(p)}(0))$, consider the following asynchronous optimized Schwarz algorithm. In process s, corresponding to the subspace $\Omega^{(s)}$, for any sequence of update indexes $\sigma(n)$ and time stamps of previous computations $\tau_q^{(s)}(n)$, $n = 1, \ldots$, satisfying conditions* (5)–(7)*,*

$$u^{(s)}(n+1) = \begin{cases} Solve \begin{cases} (\Delta - \eta)u^{(s)}(n+1) = f^{(s)} \quad on\ \Omega^{(s)}, \\ -\frac{\partial u^{(s)}(n+1)}{\partial x} + \Lambda(u^{(s)}(n+1)) = v - \frac{\partial u^{(s-1)}(\tau_{s-1}^{(s)}(n))}{\partial x} + v\Lambda(u^{(s-1)}(\tau_{s-1}^{(s)}(n))) \\ \frac{\partial u^{(s)}(n+1)}{\partial x} + \Lambda(u^{(s)}(n+v1)) = v\frac{\partial u^{(s+1)}(\tau_{s+1}^{(s)}(n))}{\partial x} + \Lambda(u^{(s+1)}(\tau_{s+1}^{(s)}(n))) \end{cases} & \text{if } s \in \sigma(n) \\ u^{(s)}(n) & \text{if } s \notin \sigma(n), \end{cases} \tag{29}$$

where, if $s = 1$, or $s = p$ one of the above boundary conditions is not present. Let $\lambda(k)$ be the Fourier transform of the (Robin type) operator Λ, for the frequency k, and assume that it satisfies conditions (17) *and* (18)*, with $e^{-\sqrt{\eta}W} < \phi < 1$. Then, the asynchronous optimized Schwarz algorithm* (29) *converges for any initial approximation $u(0)$.*

We turn back our attention towards our original goal of this section, namely, the solution of the problem (14) with an asynchronous optimized Schwarz method. The hypothesis (18) of Theorem 6 does not always hold for $\eta = 0$, and very small Fourier modes, and thus, the proof we present cannot be extended to this case, that is, in exact arithmetic. Instead, we consider the practical situation of an actual computation, where round off errors are naturally introduced in the solution of the Poisson equation, i.e., $\eta = 0$ in (29). This solution will be close to the solution of the shifted Laplacian with some small $\eta > 0$ (say of the order of the machine precision), for which our theoretical result holds.

5 Numerical results

We present numerical experiments that illustrate the performance of the proposed asynchronous optimized Schwarz method.

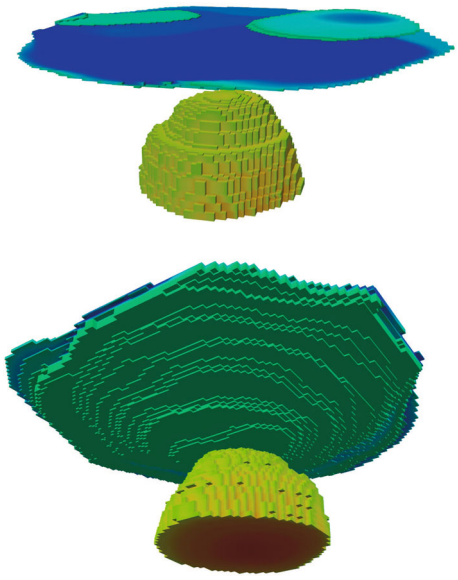

Fig. 4 Close up of the salt dome geometry

All test cases are related to the study of the Chicxulub impact crater, located underneath the town of Chicxulub, in the southeast of Mexico on the Yucatan Peninsula. One of the techniques used to study this geological formation is the gravity method, which allows to quantify differences in the Earth's gravitational field at specific locations. Detected anomalies of the gravitational field allow scientists to draw conclusions about the geological structures, and to determine the depth, density and geometry of the gravitational anomaly sources. For these numerical experiments, data acquisition have been performed on a physical domain covering the Yucatán Peninsula with an area of 250 km $\times$ 250 km and reaching 15 km in depth. From these measurements, the localization and the concentration of the salt-dome for instance can be determined with strong accuracy. The value of the density ρ, see equation below, obtained with data acquisition is shown in Fig. 4.

Using the data thus collected, one of the techniques to analyze geological formation is the method of potential, which describes the subsurface and which consists in determining mass density distribution correlated with seismic velocities. For completeness, we described this in more detail. The gravity force is the resultant of the gravitational force and the centrifugal force. The gravitational potential of a spherical density distribution is equal to $\Phi(r) = Gm/r$, with m the mass of the object, r the distance to the object and G the universal gravity constant equal to $G = 6.672 \times 10^{-11}\ \mathrm{m^3\ kg^{-1}\ s^{-2}}$. The gravitational potential at a given position x initiated by an arbitrary density distribution ρ is given by $\Phi(x) = G \int (\rho(x')/||x - x'||)dx'$, where x' represents one point position within the density distribution. In this paper, we consider only the regional scale of the gravimetry equations therefore we do not take into account the effects of the centrifugal force. As a consequence, the gravitational potential Φ of a density anomaly distribution $\delta\rho$ is given as the solution of the Poisson equation $\Delta\Phi = -4\pi G\delta\rho$, where $\delta\rho$ was obtained from field measurements. Measurements of the gravitational potential

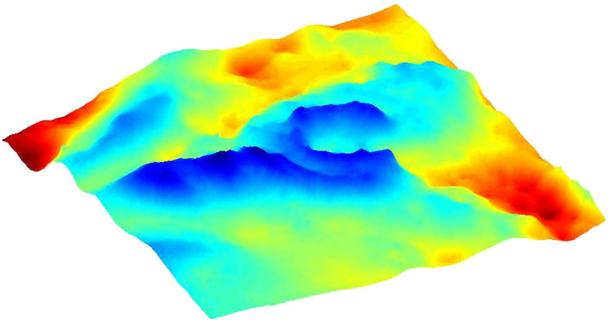

Fig. 5 Measurements of the gravitational potential in the Yucatan Peninsula

in Yucatan Peninsula is shown Fig. 5 and illustrate the strong variation of the potential in the center of this area.

The numerical solution of the gravitational potential equation required for this study was done on a parallelepiped geometric domain of dimensions 750 km $\times$ 750 km $\times$ 45 km. At the center of this geometry, the volume of dimensions 250 km $\times$ 250 km $\times$ 15 km is meshed with regular size finite elements, and outside this volume the mesh is composed of larger finite elements. One example of the finite element mesh used for the simulation is shown in Fig. 6 (top). We considered four regular meshes containing respectively 327,697 degrees of freedom (DOF), 2,491,632 DOF, 19,933,056 DOF, 146,707,292 DOF and 313,632 finite elements, 2,436,862 finite elements, 19,713,165 finite elements, 157,902,452 finite elements. Lagrange Q_1 finite elements were used for the discretization. The domain is split into subdomains and one example of the partitioning into 64 subdomains is shown in Fig. 6 (bottom).

The finite element solution on the plane 1.5 km below the surface of the domain is shown in Fig. 7 illustrating that indeed there is variation of gravity in the region which is consistent with impact of a large object.

To solve the resulting linear system, the synchronous and asynchronous Schwarz methods, with zeroth and second order optimized interface conditions, were implemented in the recently developed C++ library *Alinea* [23]. The optimized coefficients of the Schwarz algorithms are obtained using the CMA-ES algorithm, first introduced in [24]. The parallel implementation of the asynchronous Schwarz methods is quite similar to the synchronous implementation except that the asynchronous iterations and asynchronous communications are managed by a new additional layer. This new additional layer, *JACK* [25], a recently developed C++ library, is defined on top of the MPI library; the version of the MPI library used in the experiments is MPICH2 [1]. This layer allows us to use asynchronous communications between the processors and to deal with continuous requests. This new layer also contains new functionality such as the detection of the asynchronous convergence of the algorithm for a given stopping criteria. Here we use the stopping criterion developed in [2]; it is based on a leader election protocol over a tree topology, where cancellation messages are introduced in order to avoid erroneous detections. It requires to estimate two parameters. One of them defines the number of iterations that each process has to perform under local convergence before entering the leader election protocol. The second, which is

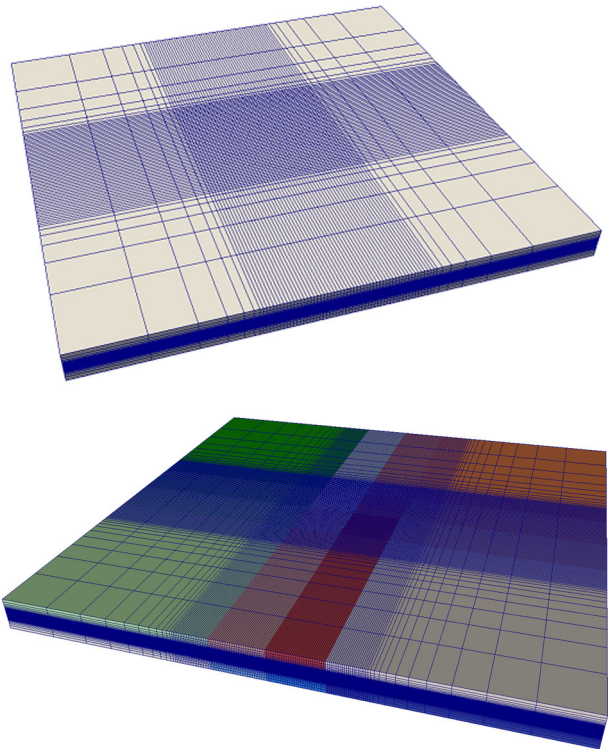

Fig. 6 Finite element mesh (*top*) and domain decomposition (*bottom*)

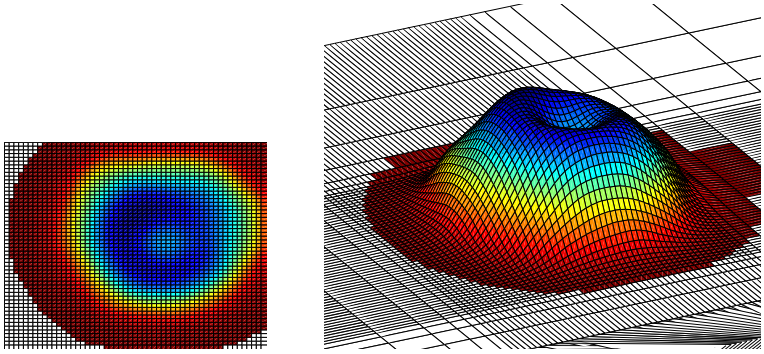

Fig. 7 Close up of the finite element solution in the center of the Yucatan Peninsula 1.5 km below the surface

critical, is the estimation of an upper bound on the maximum communication delay in the network, so that cancellation messages can always be taken into account.

We perform two sets of experiments. The first is on a fairly heterogeneous cluster composed of four nodes [Intel(R) Xeon(R) E5410, 2.33 GHz, 8 cores, RAM: 8 GB], four nodes [Intel(R) Core(TM) i7 2.80 GHz, 8 cores, RAM: 8 GB] each with graphics

Table 1 Number of iterations or updates and computational time (in seconds) for cluster of 160 cores

DOF	Subdomains	OO0	
		Iter or updates (min, mean,median,max)	Time
Synchronous			
327,697	16	848	47
327,697	32	871	39
327,697	64	973	29
Asynchronous			
327,697	16	(1133, 1421, 1486, 2164)	39
327,697	32	(845, 1298, 1287, 1715)	32
327,697	64	(889, 1112, 1204, 1604)	23

Table 2 Number of iterations or updates and computational time (in seconds) for supercomputer of 1104 processors

DOF	Subdomains	OO0		OO2	
		Iter or updates (min, max)	Time	Iter or updates (min, max)	Time
Synchronous					
2,491,632	256	1722	43.4	575	14.2
19,933,056	512	3379	777.0	938	213.9
146,707,292	1024	8331	3888.0	1850	863.3
Asynchronous					
2,491,632	256	(1030, 1917)	40.2	(309, 1334)	13.5
19,933,056	512	(2257, 4438)	590.8	(627, 2714)	175.5
146,707,292	1024	(5251, 13274)	863.2	(811, 4820)	352.4

processing units accelerator (Tesla K20c 4799 MB, GTX 570 1279 MB) and four nodes [Intel(R) Xeon(R) E5-2609, 2.10 GHz, 24 cores, RAM: 16 GB] each with graphics processing units accelerator (three of them with Quadro K4000 3071 MB, and one with Quadro K600 1023 MB), for a total of 160 cores. The interconnected network is a switched, star shaped 10 Mb/s Ethernet network. We report our computational results in Table 1, where we vary the number of subdomains using the same discretization.

The machine used for the second set of numerical experiments reported in Table 2 is fairly homogeneous; it is a Bull NOVA (NovaScale servers) supercomputer, composed of 1068 computing nodes and a few gateway, management and storage nodes. Each computing node has four quadricore processors and 48 GB of memory. Half of the nodes are equipped with "Xeon E7310 Tigerton" at 1.6 Ghz with 2×2MB of cache and the other half are equipped with "Xeon E7350 Tigerton" at 2.93 Ghz with 2×4 MB of cache. All the nodes are connected with InfiniBand (20 GB/s). As we can see, the maximum number of updates for the asynchronous version is always larger than the number of iterations in the synchronous version. Nevertheless, the computational

time for the former is lower. This illustrates that the proposed method not only is suitable for parallel computation, but that it is expected to be faster than the more traditional synchronous method. In fact, for the larger problem, the asynchronous method is faster by a factor of 4.5 in the case of OO0 interface conditions, and by a factor of almost 2.5 in the case of OO2.

6 Conclusions

In this paper an asynchronous version of the optimized Schwarz method is proposed. Here, asynchronous methods refer to parallel iterations where each process performs its task without waiting for other processes to be completed, i.e., with whatever information it has locally available and with no synchronizations with other processes.

We first prove the convergence of the asynchronous iterations for a general differential operator and optimal interface conditions. We then show the convergence for approximate optimized conditions for the Laplacian operator in practical computations. Numerical experiments conducted on the gravitational analysis in the Yucatan Peninsula illustrate the convergence properties of the asynchronous optimized Schwarz method. As reflected in the numerical experiments, despite an increase of the number of updates, the asynchronous version outperforms the synchronous version in computational time.

Acknowledgements The authors would like to express their personal appreciation of the valuable assistance given by A.K. Cheik Ahamed, G. Guillaume Gbikpi-Benissan, and B. Zhang who programmed certain parts of the codes used in our experiments. They also thank R. Martin for the data on the Chicxulub impact crater, Y. Grabovsky for a helpful conversation, and J. Garay and two anonymous referees for their questions and comments on an ealier version of this paper, which help improved our results and presentation.

References

1. Argonne National Laboratory. MPI: The Message Passing Interface.Technical report, Mathematics and Computer Science Division, ArgonneNational Laboratory (1993)
2. Bahi, J.M., ContassotVivier, S., Couturier, R.: Parallel Iterative Algorithms: From Sequential to Grid Computing. Chapman & Hall, Boca Raton (2008)
3. Bertsekas, D.P.: Distributed asynchronous computation of fixed points. Math. Program. **27**, 107–120 (1983)
4. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Prentice Hall, Englewood Cliffs (1989)
5. Chau, M., Garcia, T., Spiteri, P.: Asynchronous Schwarz methods applied to constrained mechanical structures in grid environment. Adv. Eng. Softw. **74**, 1–15 (2014)
6. Chevalier, P., Nataf, F.: Symmetrized method with optimized second-order conditions for the Helmholtz equation. In: Mandel, J., Farhat, C., Cai, C.-C. (eds.) Domain Decomposition Methods 10, Contemporary Mathematics, vol. 218, pp. 400–407. Boulder, CO (1997). American Mathematical Society, Providence, RI (1998)
7. Dolean, V., Lanteri, S., Nataf, F.: Optimized interface conditions for domain decomposition methods in fluid dynamics. Int. J. Numer. Methods Fluids **40**, 1539–1550 (2002)
8. Dolean, V., Lanteri, S., Perrussel, R.: Optimized Schwarz algorithms for solving time-harmonic Maxwell's equations discretized by a discontinuous Galerkin method. IEEE Trans. Magn. **44**, 954–957 (2008)
9. Dubois, O., Gander, M.J., Loisel, S., St-Cyr, A., Szyld, D.B.: The optimized Schwarz method with a coarse grid correction. SIAM J. Sci. Comput. **34**, A421–A458 (2012)

10. El Tarazi, M.N.: Some convergence results for asynchronous algorithms. Numer. Math. **39**, 325–340 (1982)
11. Frommer, A., Schwandt, H., Szyld, D.B.: Asynchronous weighted additive Schwarz methods. Electron. Trans. Numer. Anal. **5**, 48–61 (1997)
12. Frommer, A., Szyld, D.B.: On asynchronous iterations. J. Comput. Appl. Math. **123**, 201–216 (2000)
13. Gander, M., Magoulès, F., Nataf, F.: Optimized Schwarz methods without overlap for the Helmholtz equation. SIAM J. Sci. Comput. **24**, 38–60 (2002)
14. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**, 699–731 (2006)
15. Gander, M.J., Halpern, L.: Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems. SIAM J. Numer. Anal. **45**, 666–697 (2007)
16. Gander, M.J., Halpern, L., Magoulès, F.: An optimized Schwarz method with two-sided Robin transmission conditions for the Helmholtz equation. Int. J. Numer. Methods Fluids **55**, 163–175 (2007)
17. Gander, M.J., Loisel, S., Szyld, D.B.: An optimal block iterative method and preconditioner for banded matrices with applications to PDEs on irregular domains. SIAM J. Matrix Anal. Appl. **33**, 653–680 (2012)
18. Laitinen, E.J., Lapin, A.V., Pieskä, J.: Asynchronous domain decomposition methods for continuous casting problem. J. Comput. Appl. Math. **194**, 393–413 (2003)
19. Lions, P.L.: On the Schwarz alternating method. I. In: Glowinski, R., Golub, G.H., Meurant, G.A., Périaux, J. (eds.) First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia. SIAM (1988)
20. Maday, Y., Magoulès, F.: Absorbing interface conditions for domain decomposition methods: a general presentation. Comput. Methods Appl. Mech. Eng. **195**, 3880–3900 (2006)
21. Maday, Y., Magoulès, F.: Optimized Schwarz methods without overlap for highly heterogeneous media. Comput. Methods Appl. Mech. Eng. **196**, 1541–1553 (2007)
22. Magoulès, F.: Asynchronous Schwarz methods for peta and exascale computing. In: Topping, B.H.V., Iványi, P. (eds.) Developments in Parallel, Distributed, Grid and Cloud Computing for Engineering, Chap. 10, pp. 229–248. Saxe-Coburg, Stirlingshire (2013)
23. Magoulès, F., Ahamed, A.-K.C.: Alinea: an advanced linear algebra library for massively parallel computations on graphics processing units. Int. J. High Perform. Comput. Appl. **29**, 284–310 (2015)
24. Magoulès, F., Ahamed, A.-K.C., Putanowicz, R.: Optimized Schwarz method without overlap for the gravitational potential equation on cluster of graphics processing unit. Int. J. Comput. Math. **93**, 955–980 (2016)
25. Magoulès, F., Gbikpi-Benissan, G.: JACK: an asynchronous communication kernel for massively parallel computations. J. Supercomput. 1–20. doi:10.1007/s11227-016-1702-2
26. Magoulès, F., Roux, F.-X., Salmon, S.: Optimal discrete transmission conditions for a non-overlapping domain decomposition method for the Helmholtz equation. SIAM J. Sci. Comput. **25**, 1497–1515 (2004)
27. Magoulès, F., Roux, F.-X., Series, L.: Algebraic approximation of Dirichlet-to-Neumann maps for the equations of linear elasticity. Comput. Methods Appl. Mech. Eng. **195**, 3742–3759 (2006)
28. Magoulès, F., Szyld, D.B.: Asynchronous Optimized Schwarz for the Helmholtz Equation (2017) (in preparation)
29. Martin, V.: An optimized Schwarz waveform relaxation method for the unsteady convection diffusion equation in two dimensions. Appl. Numer. Math. **52**, 401–428 (2005)
30. Mathew, T.P.A.: Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations, vol. 61 of Lecture Notes in Computational Science and Engineering. Springer, Berlin (2008)
31. Nataf, F.: Recent developments on optimized Schwarz methods. In: Widlund, O.B., Keyes, D. (eds.) Domain Decomposition Methods in Science and Engineering XVI, vol. 55. Lecture Notes in Computational Science and Engineering, pp. 115–125. Springer, Berlin (2007)
32. Nataf, F., Rogier, F., de Sturler, E.: Optimal Interface Conditions for Domain Decomposition Methods. Technical Report 301, CMAP (École Polytechnique) (1994)
33. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York, London (1970). Reprinted by SIAM, Philadelphia (2000)
34. Quarteroni, A., Valli, A.: Domain Decomposition Methods for Partial Differential Equations. Oxford Science Publications, Oxford (1999)
35. Schwarz, H.A.: Gesammelte Mathematische Abhandlungen, vol. 2, pp. 133–143. Springer, Berlin (1890). First published in Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, vol. 15, pp. 272–286 (1870)

36. Smith, B.F., Bjørstad, P.E., Gropp, W.D.: Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press, Cambridge (1996)
37. Toselli, A., Widlund, O.B: Domain Decomposition Methods—Algorithms and Theory, vol. 34 of Series in Computational Mathematics. Springer, Berlin (2005)
38. Üresin, A., Dubois, M.: Sufficient conditions for the convergence of asynchronous iterations. Parallel Comput. **10**, 83–92 (1989)