Convergence analysis of overlapping domain decomposition preconditioners for nonlinear problems

El Mehdi Ettaouchi, Luc Giraud, Carola Kruse, Nicolas Tardieu

Convergence analysis of overlapping domain decomposition preconditioners for nonlinear problems

Ettaouchi El Mehdi, Luc Giraud, Kruse Carola, Nicolas Tardieu

RESEARCH REPORT
N° 9575
February 2025
Project-Team Concace

ISSN 0249-6399 ISRN INRIA/RR--9575--FR+ENG

Convergence analysis of overlapping domain decomposition preconditioners for nonlinear problems

Ettaouchi El Mehdi*, Luc Giraud†, Kruse Carola‡, Nicolas Tardieu*

Project-Team Concace

Research Report n° 9575 — February 2025 — 36 pages

Abstract: Numerical simulations of nonlinear partial differential equations often involve solving large nonlinear systems, for which Newton's method is widely employed due to its fast convergence near the solution. However, its performance can deteriorate in the presence of strong nonlinearities or poor initial guesses. Nonlinear overlapping domain decomposition methods, such as RASPEN [11] and Substructured RASPEN (SRASPEN) [5], have proven effective in addressing these challenges. Because SRASPEN reduces the problem size by restricting computations to a substructure, it does not update the solution outside the substructure, so that no natural initial guesses for the nonlinear local solution exists that might lead to additional inner subdomain nonlinear iterations or even prevent the local solvers to converge. In this study, we analyze the convergence of RASPEN. We show how domain decomposition improves the convergence rate of the Newton's method by highlighting the key role of the substructure on the global error contraction. Moreover, our analysis provides insight into an inexpensive modification to SRASPEN that mitigates the lack of iterations outside the substructure. The proposed variant significantly reduces computational cost while improving overall efficiency compared to existing techniques in the literature. Numerical experiments confirm the computational performance and robustness of the improved SRASPEN, establishing it as a reliable approach for solving large-scale nonlinear systems.

Key-words: Newton, nonlinear preconditioner, Additive Schwarz

* EDF - R&D Dpt. ERMES, Paris Saclay
† Inria, Inria centre at the University of Bordeaux
‡ Cerfacs, Toulouse

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Analyse de convergence des préconditionneurs de décomposition de domaine avec recouvrement pour problèmes non linéaires

Résumé : Les simulations numériques d'équations aux dérivées partielles non linéaires impliquent souvent la résolution de grands systèmes non linéaires, pour lesquels la méthode de Newton est largement utilisée en raison de sa convergence rapide près de la solution. Cependant, ses performances peuvent se détériorer en présence de fortes non-linéarités ou de mauvaises suppositions initiales. Les méthodes de décomposition de domaines non linéaires par chevauchement, telles que RASPEN [11] et Substructured RASPEN (SRASPEN) [5], se sont avérées efficaces pour relever ces défis. Comme SRASPEN réduit la taille du problème en limitant les calculs à une sous-structure, il ne met pas à jour la solution en dehors de la sous-structure, de sorte qu'il n'existe pas de suppositions initiales naturelles pour la solution locale non linéaire qui pourraient conduire à des itérations non linéaires supplémentaires dans le sous-domaine intérieur ou même empêcher les solveurs locaux de converger. Dans cette étude, nous analysons la convergence de RASPEN. Nous montrons comment la décomposition du domaine améliore le taux de convergence de la méthode de Newton en mettant en évidence le rôle clé de la sous-structure sur la contraction de l'erreur globale. En outre, notre analyse donne un aperçu d'une modification peu coûteuse de SRASPEN qui atténue le manque d'itérations en dehors de la sous-structure. La variante proposée réduit considérablement les coûts de calcul tout en améliorant l'efficacité globale par rapport aux techniques existantes dans la littérature. Des expériences numériques confirment les performances de calcul et la robustesse de la SRASPEN améliorée, l'établissant comme une approche fiable pour la résolution de systèmes non linéaires à grande échelle.

Mots-clés : Newton, préconditionneur non-linéaire, Schwarz additif

1. Introduction. In many applications, numerical simulations involve solving a nonlinear system of equations resulting from a discretisation technique (such as the finite element or finite volume methods) of the associated nonlinear partial differential equations. Generally, Newton's method is considered the first choice for solving such problems due to its quadratic convergence rate when the iterates enter the so called basin of attraction near the solution [20]. However, it can exhibit slow transient nonlinear convergence in the early stages when the system is characterised by strong nonlinearities and the initial guess is far from the basin of attraction. Acceleration techniques, often referred to as preconditioning, have to be considered to improve the robustness of the Newton iterations. Preconditioning is a general paradigm that appears in many contexts, originally introduced in numerical linear algebra to speed-up the convergence of iterative methods for the solution of linear systems or the computation of eigenpairs. The general idea is to transform the original problem into a preconditioned one that is more easily amenable to the solution by the selected numerical scheme. For large scale problems, in addition to reducing the number of iterations, the so-called preconditioner is designed to exhibit natural parallelism for enabling its implementation on parallel computing facilities. Well-known examples for the solution of linear systems arising from the discretization of PDE are the domain decomposition techniques [27, 29, 31], which will play a crucial role in the following.

In the nonlinear context, we aim to find the root u of F such that $F(u) = 0$ and preconditioning consists in applying a nonlinear operator G to F, which transforms the problem into one of two possible forms: $F(G(v)) = 0$ or $G(F(v)) = 0$, where $G(0) = 0$. The latter case $(G(F(v)) = 0)$ corresponds to left preconditioning where the new equation has the same solution u as the original problem. In the former situation of right preconditioner, $(F(G(v)) = 0)$, the solution is related to the original one by a change of variables, $u = G(v)$. In both cases, the new nonlinear equation is constructed to have improved properties to allow Newton's method to converge more efficiently. This improvement is achieved when the operator G somehow approximates F^{-1}, resulting in a transformed function that is closer to linear. As a consequence, the Jacobian of the new system is closer to the identity matrix, which makes the solution of the linear systems easier to handle at each step of Newton's method. Right nonlinear preconditioning applies a change of variable using a nonlinear transformation. In [2] is shown that applying Newton iteration to the preconditioned system essentially results in applying a Newton iteration to the original system but moving the iterate at each iteration before computing the Newton step (we refer to [2] for the technical details). Techniques based on this strategy are called nonlinear elimination [18, 23, 24, 25]. Nonlinear variants of FETI-DP [15, 16] or BDDC [10, 26] were shown to be effective for the solution of large nonlinear problems [21, 22].

In this work, we consider left preconditioning. This approach was first introduced in [3] where the overlapping nonlinear additive Schwarz method was considered to enhance the inexact Newton (IN) method [8, 9, 14, 20]. The resulting Additive Schwarz Preconditioned Inexact Newton's method (ASPIN) has since evolved into the Restricted Additive Schwarz Preconditioned Exact Newton (RASPEN) approach. RASPEN combines a restricted version of the parallel nonlinear Schwarz method with exact Newton iterations [11], improving both linear and nonlinear convergence rates due to the availability of exact Jacobian information. Other strategies involve employing advanced preconditioners, such as Dirichlet-Neumann methods, to develop the Dirichlet-Neumann Preconditioned Exact Newton (DNPEN) method [5].

In their article on the substructured variant of RASPEN (SRASPEN) [4], the authors demonstrate that the iterates of the RASPEN method, when restricted to the unknowns of the skeleton, are identical to those generated by their SRASPEN method, which is defined only on the skeleton. The key advantage of SRASPEN lies in working with lower-dimensional spaces while still requiring, like RASPEN, the resolution of nonlinear local problems within the subdomains, including the internal vertices that do not belong to the skeleton. In this work, we show that,

for the RASPEN method, the convergence to the solution is as fast at interior vertices as it is on the skeleton. This led us to develop a way to recover the interior iterates of RASPEN from those of SRASPEN. As a result, we obtain highly effective initial guesses for the nonlinear local problem resolutions required at each global nonlinear step. These initial guesses ensure that the computations at the subdomain level start within the Newton's method's basin of attraction, thereby enhancing the robustness of SRASPEN and significantly reducing its computational costs compared to approaches proposed in the literature.

We begin in Section 2 by introducing the nonlinear Schwarz method and its variants, illustrating their role as left preconditioners within the Newton's method. Section 3 presents a convergence analysis of RASPEN, establishing new bounds on both its pre-asymptotic and asymptotic convergence regimes. Our findings reveal that the convergence for the unknowns outside the skeleton is as fast as for those on the skeleton. This key observation motivates the development of a novel strategy to construct effective initial guesses for the subdomain nonlinear solves required at each SRASPEN iteration. As a result, we introduce a new SRASPEN variant that combines the main advantages of both RASPEN and SRASPEN. Finally, Section 4 presents numerical experiments comparing the performance of RASPEN and the new enhanced SRASPEN on test models, illustrating the improvements achieved.

2. Nonlinear Overlapping Schwarz Method.

2.1. Model Problem. Let us consider the following partial differential equation (PDE) on a domain Ω:

$$\begin{cases} \mathcal{D}(u) = 0 & u \in H^1(\Omega), \\ u = u_D & \text{on } \partial\Omega_D, \end{cases}$$

where $\mathcal{D}$ is a nonlinear differential operator, and u_D represents the Dirichlet boundary condition function on the domain boundary $\partial\Omega_D$. Let Ω_h denote the finite element mesh of Ω with element size h, and V_h the finite element space of dimension n associated with this mesh. We seek a solution $u_h^* \in V_h$ such that:

$$\int_\Omega \mathcal{D}(u_h^*)\ v\ dx = 0 \qquad \forall v \in V_h, \tag{2.1}$$

$$u_h^* = u_D \qquad \text{on } \partial\Omega_D. \tag{2.2}$$

This system represents the variational formulation corresponding to the original problem. Projecting Equation (2.1) onto the function basis of V_h, while incorporating the Dirichlet boundary condition (2.2), results in a nonlinear system expressed as:

$$F(u_h^*) = 0. \tag{2.3}$$

Here, F is a nonlinear function representing the finite element residual, for which we aim to find the root. For the remainder of the paper, we will omit the subscript h and denote u^* as the unique solution of (2.3) in the finite element space V.

2.2. Decomposition of the Domain. Let us begin by introducing the nonlinear overlapping Schwarz method for the solution of (2.3). Let $\left(\Omega_i^0\right)_{i\in[\![1,N]\!]}$ be a nonoverlapping partition of Ω such that:

$$\begin{cases} \Omega = \bigcup_{j=1}^{N} \Omega_j^0, \\ \Omega_j^0 \cap \Omega_i^0 = \emptyset \quad i \neq j. \end{cases}$$

For brevity and clarity we will abuse the notation and employ Ω_i^0 to denote not only a subdomain but also the mesh points defined on it. Based on this partitioning, we can define overlapping subdomains by enlarging each of them to include (w.r.t. the connectivity graph) neighboring nodes within a distance of $\delta \times h$, where δ represents the number of neighborhood levels considered. This allows us to define a subdomain Ω_i^δ with an overlap $\xi = \delta \times h$. Let V_i^0 and V_i^δ be respectively the finite element discretization spaces on Ω_i^0 and Ω_i^δ, we define respectively the prolongations $P_i^0 : V_i^0 \longrightarrow V$, $P_i^\delta : V_i^\delta \longrightarrow V$ such that:

$$\begin{cases} V = \bigoplus_{j=1}^{N} P_i^0 V_i^0, \\ V = \sum_{j=1}^{N} P_i^\delta V_i^\delta. \end{cases} \tag{2.4}$$

Given the prolongation operators P_i^0, P_i^δ, the restriction operators R_i^0, R_i^δ are defined as their respective adjoint operators that verify:

$$\begin{cases} R_i^\delta P_i^\delta = I_{V_i^\delta} \quad , \quad \sum_{i=1}^{N} D_i^\delta P_i^\delta R_i^\delta = I_V, \\ R_i^0 P_i^0 = I_{V_i^0} \quad , \quad \sum_{i=1}^{N} P_i^0 R_i^0 = I_V, \end{cases}$$

where $\left(D_i^\delta\right)_{i \in [\![1,N]\!]}$ are $V \to V$ diagonal operators forming a partition of unity. From now on, for ease of notation and reading, we will drop the superscript δ from any operator or quantity associated with the overlapping subdomains. Having decomposed V into $(V_i)_{i \in [\![1,N]\!]}$ (as in (2.4)), we can define for each element u of V a local approximation to the original function F, denoted $F_u^{(i)}$, that is given by:

$$\begin{aligned} F_u^{(i)} : \quad & V_i \longrightarrow V_i, \\ & v_i \longmapsto R_i F\left(u - P_i v_i\right). \end{aligned}$$

With the local approximations $\left(F_u^{(i)}\right)_{i \in [\![1,N]\!]}$, we seek to solve the following N subproblems:

$$\forall i \in [\![1, N]\!], \quad \text{Find } v_i \in V_i \text{ such that:} \qquad F_u^{(i)}\left(v_i\right) = 0_{V_i}. \tag{2.5}$$

Let us first discuss the dependencies of these local solutions $(v_i)_{i \in [\![1,N]\!]}$ and then their existence and uniqueness. In our notation, we indicated only the dependence on the subdomain through the index i. However, since v_i is a solution to $F_u^{(i)}$, which in turn depends on F and u, v_i will also depend on these two quantities. Thus, if we want to denote the process of finding these local solutions as a function, we can denote it as $v_i = C_i\left(F, u\right)$, where the letter C is chosen to indicate that v_i is a correction. Nevertheless, is C_i, the function that maps $\mathscr{F}\left(V\right) \times V$ to V_i, always defined for any function in $\mathscr{F}\left(V\right)$ along with any element u in V? The answer is no. Only a sufficient condition on $F_{u^*}^{(i)}$ is given in [13], which states that its differential at 0_{V_i} is nonsingular. This implies that $R_i dF\left(u^*\right) P_i$ is nonsingular, where $dF\left(u^*\right)$ denotes the differential of F at u^*. This is a local existence condition, meaning that even if it is satisfied by F, $C_i\left(F, u\right)$ exists and is unique only in an open ball neighborhood $U = \mathcal{B}\left(u^*, r^*\right)$ of the solution u^*; hence, u needs to

be in U. Whenever we use the term $C_i(F,u)$, we assume that $R_i dF(u^*) P_i$ is nonsingular and $u \in U$. By definition, $C_i(F,u)$ satisfies the following equation:

$$R_i F(u - P_i C_i(F,u)) = 0. \tag{2.6}$$

From a finite element perspective, the process of finding $C_i(F,u)$ corresponds to solving the following system:

Find $w_i \in V_i$, such that:

$$\begin{cases} \displaystyle\int_\Omega \mathcal{D}_{V_i}(w_i)\ v_i\ dx = 0 & \forall v_i \in V_i, \\ w_i = u_D & \text{on } \Gamma_i^D = \partial\Omega_i \cap \partial\Omega_D, \\ w_i = u & \text{on } \Gamma_i^{\text{int}} = \partial\Omega_i \setminus \partial\Omega_D, \end{cases} \tag{2.7}$$

where Γ_i^D is the local boundary that corresponds to the natural Dirichlet boundary $\partial\Omega_D$, Γ_i^{int} is the internal local boundary induced by the domain decomposition. The solution w_i of (2.7) is the locally corrected solution, and the local correction is simply defined as:

$$C_i(F,u) = R_i u - w_i.$$

The local corrections being defined, we define a nonlinear preconditioner referred to as $\mathcal{P}_{AS}$ representing the action of nonlinear additive Schwarz that is given by:

$$\mathcal{P}_{AS}(F,u) = u - \sum_{i=1}^{N} P_i C_i(F,u).$$

A direct way to use $\mathcal{P}_{AS}$ is in a fixed point iteration as follows:

$$u_{j+1} = \mathcal{P}_{AS}(F,u_j). \tag{2.8}$$

When $\mathcal{P}_{AS}$ is used in a fixed point iteration it is considered as a solver. However, in many cases, it will not consist of a robust solver. For that reason, $\mathcal{P}_{AS}$ is commonly considered as a preconditioner in order to help improving the convergence of other solvers known to be more robust; this will be the topic of the upcoming section. Both $(C_i)_{i\in[\![1,N]\!]}$ and $\mathcal{P}_{AS}$ depend on F. Until now, we made this dependence explicit to emphasize that the nonlinear action of the preconditioner $\mathcal{P}_{AS}$ requires knowledge of the function F itself in addition to the element where it will be applied. Thus, we write $\mathcal{P}_{AS}(F,u)$ instead of $\mathcal{P}_{AS}(F(u))$, which is valid only in the linear case. For the sake of clarity, we will keep this dependency implicit and simply write $C_i(u) = C_i(F,u)$ and $\mathcal{P}_{AS}(u) = \mathcal{P}_{AS}(F,u)$.

2.3. Schwarz methods accelerated by Newton. In general, the use of nonlinear preconditioners in fixed point iterations can lead to divergence or very slow convergence rates. To mitigate this issue, an outer solver is often introduced to accelerate convergence, with the Newton's method being a common choice. This method involves solving a new function associated with the action of the fixed point iteration induced by the preconditioner. Consequently, the original function F is replaced with a new one that depends on the preconditioner $\mathcal{P}_{AS}$ and its convergence properties. As recalled in the introduction, the first use of an outer solver for the fixed point iteration associated with $\mathcal{P}_{AS}$ used an inexact Newton's method [3]. Later, the exact Newton's method was employed along with a restricted version of the Schwarz preconditioner [11]. In the following sections, we will discuss each of these methods, providing the expressions of their functions and Jacobians.

2.3.1. Additive Schwarz Preconditioned Exact Newton (ASPEN). We now use the nonlinear additive Schwarz preconditioner $\mathcal{P}_{AS}$ in order to construct a new function denoted $\mathcal{F}_{AS}$ (which admits u^* as a solution) on which applying Newton's method would lead to a faster and robust convergence. For that, $\mathcal{F}_{AS}$ is defined by the action of the fixed point iteration induced by $\mathcal{P}_{AS}$ as follows:

$$\mathcal{F}_{AS}(u) = u - \mathcal{P}_{AS}(u) = \sum_{i=1}^{N} P_i C_i(u).$$

Hence, we are now interested in solving the new nonlinear equation:

$$\mathcal{F}_{AS}(u) = 0_V. \tag{2.9}$$

It is straightforward to prove that u^* is a solution to $\mathcal{F}_{AS}$ (Theorem 5.4 in [3]). However, the current analysis only establishes the uniqueness of u^* within a limited neighborhood. Showing global uniqueness for u^* in the context of Equation (2.9) for any overlap size remains case dependent. The approach of solving (2.9) using Newton's method is referred to as ASPEN. This method results in the following iterative process:

$$u_{j+1} = u_j - \left(d\mathcal{F}_{AS}(u_j)\right)^{-1} \mathcal{F}_{AS}(u_j), \tag{2.10}$$

where $d\mathcal{F}_{AS}$ denotes the differential operator of $\mathcal{F}_{AS}$. It is expressed in terms of local correction differentials $(dC_i)_{i\in[\![1,N]\!]}$ as follows:

$$d\mathcal{F}_{AS}(u) = \sum_{i=1}^{N} P_i dC_i(u). \tag{2.11}$$

Differentials of the corrections are analytically obtained by differentiation of Equation (2.6), leading to:

$$\begin{aligned} &R_i dF\left(u - P_i C_i(u)\right)\left(I - P_i dC_i(u)\right) = 0_{V_i}, \\ &R_i dF\left(u - P_i C_i(u)\right) P_i dC_i(u) = R_i dF\left(u - P_i C_i(u)\right). \end{aligned}$$

Since $u \in U$, the linear operator $R_i dF\left(u - P_i C_i(u)\right) P_i$ is invertible (see [13]). Denoting $dF_i : u \mapsto dF\left(u - P_i C_i(u)\right)$, we obtain:

$$dC_i(u) = \left(R_i dF_i(u) P_i\right)^{-1} R_i dF_i(u). \tag{2.12}$$

Now that we expressed all the differential operators of the corrections, the explicit expression of $d\mathcal{F}_{AS}(u)$ is achieved by replacing the expressions of $(dC_i)_{i\in[\![1,N]\!]}$ in (2.11) which leads to:

$$d\mathcal{F}_{AS}(u) = \sum_{i=1}^{N} P_i \left(R_i dF_i(u) P_i\right)^{-1} R_i dF_i(u). \tag{2.13}$$

We can see that nonlinear Schwarz preconditioning results in a Jacobian, $d\mathcal{F}_{AS}(u)$, expressed in terms of the local operators $R_i dF_i(u) P_i$, which are derived from $dF_i(u)$. These operators are subdomain-dependent not only on the restriction and prolongation operators but also on $dF_i(u)$, which corresponds to dF evaluated at a locally corrected position. This dependence on the specific points where the Jacobians are evaluated is the major difference compared to applying a linear additive Schwarz preconditioner to $dF(u)$, where, in this case, all local operators will be derived from the same global operator.

2.3.2. Restricted Additive Schwarz Preconditioned Exact Newton (RASPEN). In [11], the authors considered an exact Newton approach with a restricted variation of $\mathcal{F}_{AS}$, expressed as follows:

$$\mathcal{F}_{RAS}(u) = \sum_{i=1}^{N} P_i^0 C_i^0(u), \tag{2.14}$$

where $C_i^0 = R_i^0 P_i C_i$ represents the restricted local corrections. The proof of u^* being a root of $\mathcal{F}_{AS}$ is based on the fact that the corrections are null at u^*. This will also be the case here, because based on the expression of C_i^0, it is null when C_i is null. This ensures that $\mathcal{F}_{RAS}$ also admits u^* as a solution. The Jacobian in this case is automatically obtained from expressing $dC_i^0 = R_i^0 P_i dC_i$, hence its full expression is given by:

$$d\mathcal{F}_{RAS}(u) = \sum_{i=1}^{N} \tilde{P}_i \left(R_i dF_i(u) P_i\right)^{-1} R_i dF_i(u),$$

where $\tilde{P}_i = P_i^0 R_i^0 P_i$ is the operator consisting of setting the values on the overlap of the subdomain to zero before prolongation to the global space. With the function and the Jacobian being defined, the Newton iteration expresses:

$$u_{j+1} = u_j - \left(d\mathcal{F}_{RAS}(u_j)\right)^{-1} \mathcal{F}_{RAS}(u_j). \tag{2.15}$$

The method is referred to as RASPEN. In the remainder of this work, we will focus specifically on this method and adopt the simplified notation $\mathcal{F} = \mathcal{F}_{RAS}$ to ease notation.

2.3.3. Additive Schwarz Preconditioned Inexact Newton (ASPIN). We can also solve (2.9) with an inexact Newton's method involving the approximation:

$$\forall i \in [\![1, N]\!], \quad dF_i(u) \approx d\widehat{F}(u) = dF\left(u - \sum_{i=1}^{N} P_i C_i(u)\right).$$

This essentially removes the dependency of dF_i specifically on the i-th subdomain correction and chooses to express all the local parts of the Jacobian on the globally corrected approximation $u - \sum_{i=1}^{N} P_i C_i(u)$. This leads to an inexact Jacobian $d\widehat{\mathcal{F}}_{AS}$ expressed as:

$$d\widehat{\mathcal{F}}_{AS}(u) = \left(\sum_{i=1}^{N} P_i \left(R_i d\widehat{F}(u) P_i\right)^{-1} R_i\right) d\widehat{F}(u).$$

$d\widehat{\mathcal{F}}_{AS}(u)$ corresponds exactly to left preconditioning $d\widehat{F}(u)$ by a linear additive Schwarz preconditioner. This method, referred to as ASPIN, was introduced in [3]. The Jacobian approximation in this case does not lead to any computational savings; it mainly aims to recover the usual structure of the linear Schwarz preconditioning. Here, the point of evaluation involves the aggregation of all the corrections but this has no additional cost since the term $\sum_{i=1}^{N} P_i C_i(u)$ is computed once as it corresponds to $\mathcal{F}_{AS}(u)$.

3. Convergence Analysis. It has already been shown in [4] that the RASPEN preconditioned function, $\mathcal{F}$, can be formulated on a smaller subspace of V. This subspace is defined using the mesh connectivity table, where the nodes adjacent to each overlapping subdomain are first identified. By concatenating these subdomain neighboring nodes, we obtain what is referred to as the skeleton of the volume. As shown in [4], if we restrict $\mathcal{F}$ to this skeleton and solve the

corresponding low-dimensional problem, we expect to obtain the same iterates on the skeleton as we would have by solving the full problem on the volume through $\mathcal{F}$. This approach is known as SRASPEN (Substructured RASPEN), a term that designates the skeleton as a substructure of the global structure, which is the volume. However, in SRASPEN, since we operate only on the skeleton, information about the rest of the volume is not computed during the iterations. Although these values are not required to reach the solution, having a good approximation of them would provide an effective initial guess for our nonlinear subdomain solvers. This initial guess, in turn, helps compute the corrections $(C_i)_{1\leq i\leq N}$ on each subdomain. As Newton's method is commonly used for solving subdomain problems, an initial guess that is close to the solution is crucial—not only to ensure convergence but also to achieve the solution in few iterations. Because the RASPEN iterates are defined on the full subdomains, they can be used as an initial guess for solving the nonlinear local solution, allowing fewer nonlinear iterations than what is required for SRASPEN, which has only the current iterate defined on the skeleton and no guess for the internal unknowns. Hence, the aim of this section is to show that in RASPEN, the approximation on the rest of the volume is at the same level of precision as on the skeleton, which justifies its ability to accelerate the nonlinear local convergence. To show this, we shall establish some inequalities where the error on the full volume will be bounded by the error on the skeleton, both in the asymptotic and the pre-asymptotic phase. At the end of this section, we will propose an additional step to add in SRASPEN in order to recover exactly the same iterates as in RASPEN on the rest of the volume. In that case, SRASPEN will benefit from solving a substructured linear system at each Newton's step while converging in the same number of nonlinear local iterations as RASPEN.

We shall consider the space V equipped with an inner product $\langle .,. \rangle$ and denote by $\|.\|$ its induced norm. For example, we can take the H^1 or L^2 inner product. As we mentioned earlier, the skeleton is composed of the neighboring nodes of each subdomain. Since these subdomain neighboring nodes are not part of the subdomain but impact the correction associated with it, we shall call them "ghost nodes". Thus, the subspace associated with the skeleton will be referred to as the ghost subspace. Up until this point, this subspace was only defined geometrically through the mesh in [4]. However, the rationale behind this geometrical choice was to identify only the nodes on which there is a nonlinear dependence. Hence, we can reach this subspace in an algebraic way by first defining, for each subdomain, a subspace where the dependence is linear and using the sum of their orthogonal complements to characterize the subspace of nonlinear dependence, that is the ghost subspace. We introduce the subspace L_i of directions along which local corrections evolve linearly within their associated nonoverlapping subdomain:

$$L_i = \left\{ w \in V \quad \middle| \quad \forall v \in U, \quad \left(R_i^0 - dC_i^0\left(v\right)\right) w = 0_{V_i^0} \right\}. \tag{3.1}$$

Since, from a geometrical perspective, we have identified the subdomain neighboring nodes as the source of nonlinearities, **can we prove that their corresponding directions are the only ones missing from L_i?**

First, we can see from the expression of $dC_i\left(v\right)$ in (2.12) that $dC_i\left(v\right) P_i = I_{V_i}$, which directly implies $\mathrm{Im}(P_i) \subset L_i$. Hence, the directions w associated with the subdomain nodes are in L_i. On the other hand, the complement subspace of $\mathrm{Im}(P_i)$, that is $\mathrm{Im}(I_V - P_i R_i)$, consists of directions w associated with nodes outside the overlapping subdomain, which satisfy $R_i^0 w = 0_{V_i^0}$. Consequently, the condition for them to be in L_i is $dC_i^0(v)w = 0_{V_i^0}$ for all $v \in U$. Given that $w \in \mathrm{Im}(I_V - P_i R_i)$, we are interested in the null space of $dC_i^0(v)(I_V - P_i R_i)$, which is equal to $\left(R_i dF_i(v) P_i\right)^{-1} R_i dF_i(v)(I_V - P_i R_i)$. As a consequence, it corresponds to the null space of the operator $R_i dF_i(v)(I_V - P_i R_i)$. If we closely examine the latter, it represents the off-diagonal of the subdomain part of $dF_i(v)$, characterized by zeros in the columns representing nodes that are

not neighbors to the subdomain. Consequently, the only directions that we cannot definitively include in L_i are the ones associated with the subdomain ghost nodes. Thus, we define $V_i^{(g)}$, the local ghost subspace that characterizes these nodes, as the orthogonal complement of L_i:

$$V_i^{(g)} = L_i^{\perp}.$$

We define the global ghost subspace $V^{(g)}$ as the sum of the local ghost subspaces:

$$V^{(g)} = \sum_{i=1}^{N} V_i^{(g)}. \tag{3.2}$$

This indicates that the nonlinearities are restricted to $V^{(g)}$, which generally has a much smaller dimension than V. By construction, the dimension of this subspace increases with respect to the number of subdomains. We shall now assume that $\dim V^{(g)} < n$ and consider the following direct decomposition of the full space V:

$$V = V^{(g)} \oplus {V^{(g)}}^{\perp},$$

where ${V^{(g)}}^{\perp}$ denotes the orthogonal complement subspace of $V^{(g)}$. We denote by $\Pi_g : V \longrightarrow V^{(g)}$ the orthogonal projection onto $V^{(g)}$. We also define the characteristic affine subspace $V_{*,\perp}^{(g)}$ centered on the exact component of u^* in $V^{(g)}$ and colinear to ${V^{(g)}}^{\perp}$:

$$V_{*,\perp}^{(g)} = \Pi_g\left(u^*\right) + {V^{(g)}}^{\perp}. \tag{3.3}$$

Hence, we have defined all the subspaces that will characterize the behavior of $\mathcal{F}$. In the following section, we will show, through lemmas and theorems, some properties of the action of the function $\mathcal{F}$ and its Jacobian on directions of the subspace ${V^{(g)}}^{\perp}$, showing the impact of these properties on the Newton iterator, and consequently, the convergence factor in both the pre-asymptotic and asymptotic phases. In both cases, the global error of the Newton approximation at a given step will be bounded by proportions of the error of the previous approximation restricted to $V^{(g)}$, which is typically much smaller than the error on the entire space V.

Before proceeding to the next section, we clarify that while the existence of the local corrections $\left(C_i\left(u\right)\right)_{1\leq i\leq N}$ depends only on the component of u in $V^{(g)}$, and the domain of definition of the RASPEN function should, in principle, include the entire subspace ${V^{(g)}}^{\perp}$, we will keep our domain of definition restricted to the ball U. This restriction accounts for the fact that the nonlinear solver in each subdomain exhibits only local convergence: convergence occurs only if the initial guess is sufficiently close to the solution.

3.1. RASPEN convergence analysis. With all the necessary tools now available, we will analyze the convergence of the RASPEN method and we start by establishing a characterization of the action of $d\mathcal{F}$ on elements of ${V^{(g)}}^{\perp}$ as shown in the following Lemma:

LEMMA 3.1. *Let $u \in V$, if $u \in {V^{(g)}}^{\perp}$, then:*

$$\forall v \in U, \quad d\mathcal{F}\left(v\right) u = u.$$

That is, u is a fixed point for the Jacobian $d\mathcal{F}\left(v\right)$ at any element v of the ball $U = \mathcal{B}\left(u^*, r^*\right)$.

Proof. Let $u, v \in V, U$

$$\begin{aligned} u \in V^{(g)^\perp} &\implies u \in \left(\sum_{i=1}^{N} V_i^{(g)}\right)^\perp \\ &\implies u \in V_i^{(g)^\perp} = L_i && \forall i \in [\![1, N]\!] \\ &\implies dC_i^0(v)\, u = R_i^0 u \quad \text{(by definition of } L_i \text{ in (3.1))} && \forall i \in [\![1, N]\!] \qquad (3.4) \\ &\implies \left(\sum_{i=1}^{N} P_i^0 dC_i^0(v)\right) u = \left(\sum_{i=1}^{N} P_i^0 R_i^0\right) u \end{aligned}$$

Hence, we obtain the final result:

$$\forall v \in U \quad d\mathcal{F}(v)\, u = u. \quad \square$$

The result of Lemma 3.1 establishes an important property of the subspace $V^{(g)^\perp}$ in regards of the action of $d\mathcal{F}$. It helps us to characterize the expression of the local corrections as follows:

LEMMA 3.2. *Let $u, w \in U$, where $(u - w) \in V^{(g)^\perp}$. Then, we have:*

$$C_i^0(u) = C_i^0(w) + R_i^0(u - w) \qquad \forall i \in [\![1, N]\!], \qquad (3.5)$$

and consequently:

$$\mathcal{F}(u) = \mathcal{F}(w) + u - w. \qquad (3.6)$$

Proof. Let $u, w \in U$ be such that:

$$(u - w) \in V^{(g)^\perp} = 0_V \overset{(3.4)}{\implies} dC_i^0(v)(u - w) = R_i^0(u - w) \qquad \forall v \in U, \quad \forall i \in [\![1, N]\!].$$

By choosing v as a convex combination of u and v such that $v = w + t(u - w)$, $t \in [0, 1]$, inside the ball U, we have:

$$dC_i^0(w + t(u - w))(u - w) = R_i^0(u - w) \qquad \forall t \in [0, 1], \quad \forall i \in [\![1, N]\!].$$

Denoting the function $\Psi_i : t \mapsto C_i^0(w + t(u - w))$, we have:

$$\begin{aligned} \Psi_i'(t) &= dC_i^0(w + t(u - w))(u - w) && \forall t \in [0, 1], \\ \Psi_i'(t) &= R_i^0(u - w) && \forall t \in [0, 1]. \qquad (3.7) \end{aligned}$$

Equation (3.7) itself justifies the integrability of all components of the function Ψ_i' since each component remains constant across the interval $[0, 1]$, equaling the corresponding component of $R_i^0(u - w)$. Consequently, we can safely integrate both sides of Equation (3.7) as follows:

$$\begin{aligned} \int_0^1 \Psi_i'(t)\, dt &= \int_0^1 R_i^0(u - w)\, dt && \forall i \in [\![1, N]\!], \\ \Psi(1) - \Psi(0) &= R_i^0(u - w) && \forall i \in [\![1, N]\!], \\ C_i^0(u) - C_i^0(w) &= R_i^0(u - w) && \forall i \in [\![1, N]\!], \\ C_i^0(u) &= C_i^0(w) + R_i^0(u - w) && \forall i \in [\![1, N]\!]. \end{aligned}$$

Hence, the expressions of the corrections and the expression of $\mathcal{F}$ are simply obtained by prolongation and summing over the subdomains:

$$\sum_{i=1}^{N} P_i^0 C_i^0 (u) = \sum_{i=1}^{N} P_i^0 C_i^0 (w) + \left(\sum_{i=1}^{N} P_i^0 R_i^0 \right) (u - w) .$$

This concludes the proof :

$$\mathcal{F}(u) = \mathcal{F}(w) + u - w. \square$$

Lemma 3.2 has various consequences that are summarized in the following corollary:

COROLLARY 3.3. *The following results are derived directly from Lemma 3.2:*

$$\forall u \in V_{*,\perp}^{(g)} \cap U \quad \mathcal{F}(u) = u - u^*, \tag{3.8}$$

$$\forall u \in U,\ \forall v \in V^{(g)^\perp} s.t.\ (u+v) \in U \quad \mathcal{F}(u+v) = \mathcal{F}(u) + v, \tag{3.9}$$

$$\forall u \in U,\ \forall v \in V^{(g)^\perp} s.t.\ (u+v) \in U \quad d\mathcal{F}(u+v) = d\mathcal{F}(u) . \tag{3.10}$$

Proof. Let $u \in V_{*,\perp}^{(g)} \cap U$, then by definition of $V_{*,\perp}^{(g)}$ in (3.3), we have $\Pi_g(u) = \Pi_g(u^*)$ and thus from Lemma 3.2 replacing w by u^* we get the result (3.8):

$$\begin{aligned} \mathcal{F}(u) &= \mathcal{F}(u^*) + u - u^*, \\ &= u - u^*. \end{aligned}$$

In the same manner, we obtain Equation (3.9) due to $\Pi_g(v) = 0_V, \quad \forall v \in V^{(g)^\perp}$. Equation (3.10) is obtained by differentiating (3.9) with respect to u. $\square$

Let us now consider Φ the mapping that describes the $(j+1)$-th Newton iteration defined in Equation (2.15):

$$\Phi(u) = u - (d\mathcal{F}(u))^{-1} \mathcal{F}(u) . \tag{3.11}$$

We also define the convex subset D_g given by:

$$D_g = V_{*,\perp}^{(g)} \cap U. \tag{3.12}$$

Since Φ is a Newton's method iterator, we know that $d\Phi(u^*) = 0$ which ensures that close enough to the solution and under some regularity assumptions, the convergence order will be at least two (see Theorem **10.1.7** in [28]). However, when the initial guess is sufficiently far from the solution, the convergence rate might be slow at the pre-asymptotic phase before the basin of attraction is reached. For this reason, we will first establish a first order result where the mapping Φ is a contraction and try to link for any iterate $u^{(k)}$ generated by Φ, the global error on V, that is $\|u^{(k)} - u^*\|$, to the error restricted on $V^{(g)}$, that is $\|\Pi_g \left(u^{(k)} - u^*\right)\|$. For the remainder of this work, we define the mapping $\mathcal{V}_*^{(g)}$ that gives for any $u \in U$ its closest element on D_g. It is defined by:

$$\begin{aligned} \mathcal{V}_*^{(g)} : U &\longrightarrow D_g, \\ u &\longmapsto u - \Pi_g (u - u^*) . \end{aligned} \tag{3.13}$$

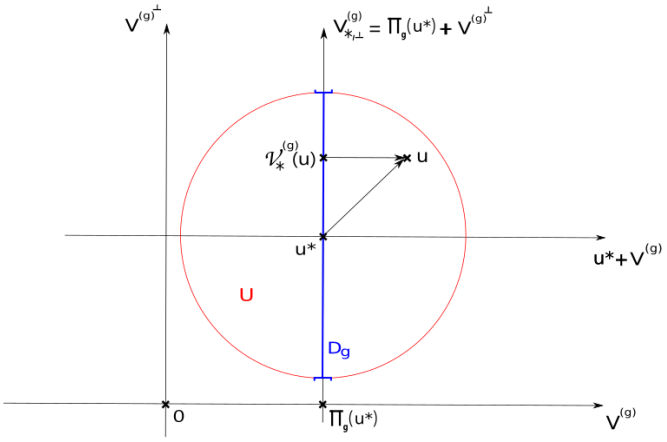


Fig. 3.1: An illustration on $\mathbb{R}^2$ of the distance between u and $\mathcal{V}_*^{(g)}(u)$ and of all subspaces and the subsets involved in our convergence analysis.

In Figure (3.1), the distance between u and $\mathcal{V}_*^{(g)}(u)$ is illustrated in $\mathbb{R}^2$, alongside the subspaces and subsets introduced earlier. Consequently, the norm of the restricted error $\|\Pi_g(u-u^*)\|$ will be referred to as the distance between u and $\mathcal{V}_*^{(g)}(u)$: $\|u-\mathcal{V}_*^{(g)}(u)\|$. We now proceed to establish a characterization of the derivatives of Φ at any order, as described below:

THEOREM 3.4. *Let $(v,w)\in U^2$ with $(v-w)\in V^{(g)^\perp}$, and let $h\in V$ be such that $v+h$ and $w+h$ remain in U. If Φ is p-times Fréchet-differentiable in U, then we have:*

$$\forall k\in[\![0,p]\!],\quad d^{(k)}\Phi(v+h)=d^{(k)}\Phi(w+h),$$

where $d^{(k)}\Phi$ denotes the k-th derivative of Φ. In particular, this result holds for any $(v,w)\in D_g^2$.

Proof. Let us consider $k\in[\![0,p]\!]$, v and w in U and h any direction of V where $\|h\|$ is small enough so that $v+h$ and $w+h$ stay in U. Assuming that $(v-w)\in V^{(g)^\perp}$, we have from Equations (3.9) & (3.10) of Corollary 3.3, the following equations:

$$\begin{cases}\mathcal{F}\left(w+h+(v-w)\right)=\mathcal{F}\left(w+h\right)+v-w,\\ \\ d\mathcal{F}\left(w+h+(v-w)\right)=d\mathcal{F}\left(w+h\right).\end{cases}\tag{3.14}$$

Also from Lemma 3.1 we have:

$$\left(d\mathcal{F}\left(w+h\right)\right)^{-1}(v-w)=v-w.\tag{3.15}$$

Developing the expression of Φ, leads to the following:

$$\begin{aligned}\Phi\left(v+h\right)&=v+h-\left(d\mathcal{F}\left(v+h\right)\right)^{-1}\mathcal{F}\left(v+h\right),\\&=v+h-\left(d\mathcal{F}\left(w+h+(v-w)\right)\right)^{-1}\mathcal{F}\left(w+h+(v-w)\right),\\&=v+h-\left(d\mathcal{F}\left(w+h\right)\right)^{-1}\mathcal{F}\left(w+h\right)-\left(d\mathcal{F}\left(w+h\right)\right)^{-1}(v-w).\end{aligned}$$

Injecting the Equation (3.15) leads us to the zero order result:

$$\Phi\left(v+h\right)=\Phi\left(w+h\right). \tag{3.16}$$

If $p>0$, and since the zero order result holds for any direction h, then k successive differentiations with respect to h of Equation (3.16) lead to the final result:

$$d^{(k)}\Phi\left(v+h\right)=d^{(k)}\Phi\left(w+h\right).$$

Particularly, for any $(v,w)\in D_g^2$, we have $(v-w)\in V^{(g)^\perp}$ and consequently the result. □

This theorem proves that the derivatives of Φ at any order remain the same for any translation inside the ball U parallel to the subspace $V^{(g)^\perp}$. This approach allows us to study the contraction of the mapping Φ by considering the distance to the subset D_g rather than directly to the fixed point u^*. The following theorem addresses this idea, suggesting that the distance from the image of any element u under the mapping Φ to the fixed point u^* is bounded by the distance of u to the subset D_g, specifically $\|u-\mathcal{V}_*^{(g)}(u)\|$, rather than by $\|u-u^*\|$. Consequently, in a region sufficiently close to the solution, where the operator norm of $d\Phi$ remains strictly less than one, Φ acts as a contraction, drawing the approximation $\Phi(u)$ closer to u^* than u is to D_g, scaled by the contraction factor. Thus, the contraction of the current error, when restricted to $V^{(g)}$, provides a bound on the error of the next approximation relative to the exact solution over the global finite element space V. This means that in at most one step, the error on the volume and particularly on the subdomains interior will be better than what we had only on the skeleton in the previous iteration. This feature is what distinguishes RASPEN from classical Newton and serves as a strong motivation to recover the values over the rest of the volume while solving only on the skeleton, because they are very precise and will allow speeding-up the nonlinear local solves. To sum up, the RASPEN iteration consists of a two-stage error reduction: the first stage replaces the distance to u^* with the distance to D_g, and the second stage applies the contraction factor to this modified distance.

THEOREM 3.5. *If Φ is continuously Fréchet-differentiable on U, then for any factor $0<c<1$, there exists $r_c>0$ such that for any $u\in U$ with $\|u-\mathcal{V}_*^{(g)}(u)\|<r_c$, we have:*

$$\|\Phi\left(u\right)-u^*\|\leq c\,\|u-\mathcal{V}_*^{(g)}\left(u\right)\|=c\,\|\Pi_g\left(u-u^*\right)\|. \tag{3.17}$$

Proof. Let $u\in U$ and $0<c<1$, we know that $d\Phi\left(u^*\right)=0$. Hence, by continuity of $d\Phi$ on U, It exists a $r_c>0$ defined as follows:

$$r_c=sup\left\{\quad 0<r<r^*\quad\middle|\quad \underset{v\in\bar{\mathcal{B}}(u^*,r)}{max}\|d\Phi\left(v\right)\|\leq c\quad\right\}. \tag{3.18}$$

Let us now assume that $\|u-\mathcal{V}_*^{(g)}\left(u\right)\|<r_c$, then by definition of r_c there exists an intermediate value $\|u-\mathcal{V}_*^{(g)}\left(u\right)\|<r_c^{int}\leq r_c$ such that:

$$\underset{v\in\bar{\mathcal{B}}(u^*,r_c^{int})}{max}\|d\Phi\left(v\right)\|\leq c.$$

Let $t\in[0,1]$, by applying Theorem 3.4 for $k=1$, $v=\mathcal{V}_*^{(g)}\left(u\right), w=u^*$ and $h=t\left(u-\mathcal{V}_*^{(g)}\left(u\right)\right)$,

we have:

$$\begin{aligned}
&\left\|d\Phi\left(\mathcal{V}_*^{(g)}(u)+t\left(u-\mathcal{V}_*^{(g)}(u)\right)\right)\right\| = \left\|d\Phi\left(u^*+t\left(u-\mathcal{V}_*^{(g)}(u)\right)\right)\right\|,\\
\Longrightarrow &\max_{v\in\left[\mathcal{V}_*^{(g)}(u),u\right]}\|d\Phi(v)\| \leq \max_{v\in\bar{\mathcal{B}}(u^*,r_c^{int})}\|d\Phi(v)\|, \quad \left(\text{with } \left[u^*,u^*+\left(u-\mathcal{V}_*^{(g)}(u)\right)\right]\subset\bar{\mathcal{B}}\left(u^*,r_c^{int}\right)\right),\\
\Longrightarrow &\max_{v\in\left[\mathcal{V}_*^{(g)}(u),u\right]}\|d\Phi(v)\| \leq c.
\end{aligned}$$

Using the inequality of the mean-value theorem on the segment $\left[\mathcal{V}_*^{(g)}(u),u\right]$, we obtain:

$$\begin{aligned}
&\left\|\Phi(u)-\Phi\left(\mathcal{V}_*^{(g)}(u)\right)\right\| \leq \max_{v\in\left[\mathcal{V}_*^{(g)}(u),u\right]}\|d\Phi(v)\|\ \|u-\mathcal{V}_*^{(g)}(u)\|,\\
&\left\|\Phi(u)-\Phi\left(\mathcal{V}_*^{(g)}(u)\right)\right\| \leq c\ \|u-\mathcal{V}_*^{(g)}(u)\|.
\end{aligned}$$

$\mathcal{V}_*^{(g)}(u)$ is in D_g, then from the result of Theorem 3.4 we have $\Phi\left(\mathcal{V}_*^{(g)}(u)\right)=\Phi(u^*)=u^*$. Thus, we are able to reach the inequality of Theorem 3.5:

$$\|\Phi(u)-u^*\| \leq c\ \|u-\mathcal{V}_*^{(g)}(u)\|.\square$$

Remark 1. *We show below that the maximum of $\|d\Phi(v)\|$ on $\bar{\mathcal{B}}(u^*,r)$ will always be reached in the affine subspace $u^*+V^{(g)}$. We can prove that for any v of the ball $\bar{\mathcal{B}}(u^*,r)$, there exists a v_g in the same ball such that $(v_g-u^*)\in V^{(g)}$ and $d\Phi(v_g)=d\Phi(v)$:*

$$\begin{aligned}
d\Phi(v) &= d\Phi\left(\mathcal{V}_*^{(g)}(v)+\left(v-\mathcal{V}_*^{(g)}(v)\right)\right),\\
&= d\Phi\left(u^*+\left(v-\mathcal{V}_*^{(g)}(v)\right)\right), \quad (\textit{ using Theorem 3.4 }),\\
&= d\Phi(v_g),
\end{aligned}$$

where $v_g-u^*=v-\mathcal{V}_*^{(g)}(v)\in V^{(g)}$. In other words, it is only the part of $u^*+V^{(g)}$ within the ball $\bar{\mathcal{B}}(u^*,r)$ controls the quality of the contraction (i.e. controls $\|d\Phi\|$).

Remark 2. *Let us define for any $u\neq u^*$ the factor $\gamma(u)=\frac{\|u-\mathcal{V}_*^{(g)}(u)\|}{\|u-u^*\|}\leq 1$, which represents, in a certain way, the inclination of $u-u^*$ from $V_{*,\perp}^{(g)}$, and by considering a sequence $\left(u^{(k)}\right)_{k\in\mathbb{N}}$ generated by the iterative process Φ converging to u^* with $\left\|u^0-\mathcal{V}_*^{(g)}(u^0)\right\|<r_c$. Then, we have the following relationship:*

$$\forall k\in\mathbb{N} \qquad \|u^{(k+1)}-u^*\| \leq c\ \gamma_k\|u^{(k)}-u^*\|,$$

where $\gamma_k=\gamma(u^{(k)})$. Consequently, it can be noted that the convergence is accelerated by the factors $(\gamma_k)_{k\in\mathbb{N}}$. However, these factors may attain a value of one, resulting in some iterations not being sped-up. Nevertheless, $\gamma_k=1$ corresponds to $\|u^{(k)}-\mathcal{V}_*^{(g)}(u^{(k)})\|=\|u^{(k)}-u^*\|$ meaning that the error is only on the subspace $V^{(g)}$ and $(I_V-\Pi_g)(u-u^*)=0$. In practice, since the error on $V^{(g)^\perp}$ is controlled by the error on $V^{(g)}$, $\|u^{(k)}-\mathcal{V}_*^{(g)}(u^{(k)})\|$, in many cases, the latter generates a non null error on $V^{(g)^\perp}$, such that $\|(I_V-\Pi_g)\left(u^{(k)}-u^*\right)\|>0$ which prevents γ_k from reaching a value of one.

The next theorem will present the asymptotic version of Theorem 3.5. When the approximate solution u is sufficiently close to the exact solution, it enters the Newton's basin of attraction. In this phase, we can still have the two-stage reduction described in Theorem 3.5, where the square of the distance to u^* is replaced by the square of the distance to D_g. This is more effective because the reduction factor in that case will be squared. Consequently, the resulting distance, weighted by a coefficient that depends on the operator norm of the second derivative at u^*, will bound the error of the next approximation over the global finite element space V. Moreover, we will show that in the basin of attraction and under a condition on the second derivative of Φ at u^*, the distance of $\Phi(u)$ to u^* on $V^{(g)^\perp}$,*i.e.*, $\|(I_V - \Pi_g)(\Phi(u) - u^*)\|$, will not vanish maintaining a minimum ratio of $\left\|u - \mathcal{V}_*^{(g)}(u)\right\|^2$.

THEOREM 3.6. *If Φ is continuously Fréchet-differentiable on U, and twice Fréchet-differentiable at u^*. Then, it exists a $\beta > 0$ and $r > 0$ such that:*

$$\|\Phi(u) - u^*\| \leq \beta \left\|u - \mathcal{V}_*^{(g)}(u)\right\|^2 \quad \forall u \in U \ s.t \quad \left\|u - \mathcal{V}_*^{(g)}(u)\right\| < r. \tag{3.19}$$

If in addition, we have:

$$(I_V - \Pi_g)\, d^{(2)}\Phi(u^*)(h)(h) \neq 0 \qquad \forall h \in V^{(g)} s.t.\ h \neq 0. \tag{3.20}$$

Then, it exists $\alpha \in]0, \beta[$, such that:

$$\|(I_V - \Pi_g)(\Phi(u) - u^*)\| \geq \alpha \left\|u - \mathcal{V}_*^{(g)}(u)\right\|^2 \quad \forall u \in U, \ \left\|u - \mathcal{V}_*^{(g)}(u)\right\| < r. \tag{3.21}$$

Proof. A general result is proven in Theorem **10.1.7** of [28]. In the same spirit, we will use the properties of the RASPEN mapping to derive the specific result of Theorem 3.6.

Let $\epsilon > 0$ and H be the following error function:

$$H(u) = d\Phi(u) - d\Phi(u^*) - d^{(2)}\Phi(u^*)(u - u^*).$$

By definition of $d^{(2)}\Phi(u^*)$, we know that it exists a $r_\epsilon > 0$ such that:

$$\|H(v)\| \leq \epsilon \|v - u^*\|, \qquad \forall v \in \mathcal{B}(u^*, r_\epsilon). \tag{3.22}$$

We shall now consider $u \in U$ such that $\left\|u - \mathcal{V}_*^{(g)}(u)\right\| < r_\epsilon$ and start developing the expression of $\|\Phi(u) - u^*\|$ in the subsequent manner:

$$\begin{aligned}
\|\Phi(u) - u^*\| &= \left\|\Phi(u) - \Phi\left(\mathcal{V}_*^{(g)}(u)\right)\right\|, \\
&\quad \text{(Using } \Phi\left(\mathcal{V}_*^{(g)}(u)\right) = \Phi(u^*) = u^* \text{ from Theorem 3.4)} \\
&= \left\|\int_0^1 d\Phi\left(\mathcal{V}_*^{(g)}(u) + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right)\left(u - \mathcal{V}_*^{(g)}(u)\right) dt\right\|, \\
&\quad \text{(Using the same technique in the proof of Lemma 3.2 through the function } \Psi) \\
&= \left\|\int_0^1 \left[d\Phi\left(\mathcal{V}_*^{(g)}(u) + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right) - d\Phi(u^*)\right]\left(u - \mathcal{V}_*^{(g)}(u)\right) dt\right\|, \\
&\quad (\text{ Because } d\Phi(u^*) = 0_{V \to V}).
\end{aligned}$$

Given that both u^* and $\mathcal{V}_*^{(g)}(u)$ are in D_g, based on Theorem 3.4 with $h = t\left(u - \mathcal{V}_*^{(g)}(u)\right)$, we find that $d\Phi\left(\mathcal{V}_*^{(g)}(u) + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right) = d\Phi\left(u^* + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right)$. Consequently, $\|\Phi(u) - u^*\|$ writes:

$$\begin{aligned}
\|\Phi(u) - u^*\| &= \left\| \int_0^1 \left[d\Phi\left(u^* + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right) - d\Phi(u^*) \right] \left(u - \mathcal{V}_*^{(g)}(u)\right) dt \right\|, \\
&= \left\| \int_0^1 \left[t\, d^{(2)}\Phi(u^*)\left(u - \mathcal{V}_*^{(g)}(u)\right) + H\left(u^* + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right) \right] \left(u - \mathcal{V}_*^{(g)}(u)\right) dt \right\|, \\
&\leq \int_0^1 t \left\| d^{(2)}\Phi(u^*)\left(u - \mathcal{V}_*^{(g)}(u)\right)\left(u - \mathcal{V}_*^{(g)}(u)\right) \right\| + \\
&\qquad \left\| H\left(u^* + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right) \right\| \left\| u - \mathcal{V}_*^{(g)}(u) \right\| dt.
\end{aligned} \tag{3.23}$$

Since $\left\| u - \mathcal{V}_*^{(g)}(u) \right\| < r_\epsilon$ then $u^* + t\left(u - \mathcal{V}_*^{(g)}(u)\right)$ is in the ball $\mathcal{B}(u^*, r_\epsilon)$, hence from Equation (3.22) we have:

$$\left\| H\left(u^* + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right) \right\| \leq t\epsilon \left\| u - \mathcal{V}_*^{(g)}(u) \right\|.$$

Injecting this result in the expression of the current upper bound of $\|\Phi(u) - u^*\|$, leads us to:

$$\|\Phi(u) - u^*\| \leq \int_0^1 t \left\| d^{(2)}\Phi(u^*)\left(u - \mathcal{V}_*^{(g)}(u)\right)\left(u - \mathcal{V}_*^{(g)}(u)\right) \right\| + t\epsilon \left\| u - \mathcal{V}_*^{(g)}(u) \right\|^2 dt, \tag{3.24}$$

$$\|\Phi(u) - u^*\| \leq \frac{1}{2}\left(\left\| d^{(2)}\Phi(u^*) \right\| + \epsilon \right) \left\| u - \mathcal{V}_*^{(g)}(u) \right\|^2. \tag{3.25}$$

We established a flexible upper bound that depends on ϵ, and by taking any value of this ϵ the inequality (3.19) is proven. Now, assuming the condition (3.20) and going back to Equation

(3.23), but considering this time $\|(I_V - \Pi_g)(\Phi(u) - u^*)\|$, we have:

$$\begin{aligned}
&\|(I_V - \Pi_g)(\Phi(u) - u^*)\| \\
&= \left\| \int_0^1 \left[t\,(I_V - \Pi_g)\, d^{(2)}\Phi(u^*)\left(u - \mathcal{V}_*^{(g)}(u)\right) + (I_V - \Pi_g)\, H\left(u^* + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right)\right] \right. \\
&\qquad \left. \left(u - \mathcal{V}_*^{(g)}(u)\right) dt \right\|, \\
&\geq \int_0^1 t\ dt \left\| (I_V - \Pi_g)\, d^{(2)}\Phi(u^*)\left(u - \mathcal{V}_*^{(g)}(u)\right)\left(u - \mathcal{V}_*^{(g)}(u)\right)\right\| \\
&\qquad - \left\| \int_0^1 H\left(u^* + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right)\left(u - \mathcal{V}_*^{(g)}(u)\right) dt \right\|, \\
&\geq \frac{1}{2}\left\| (I_V - \Pi_g)\, d^{(2)}\Phi(u^*)\left(u - \mathcal{V}_*^{(g)}(u)\right)\left(u - \mathcal{V}_*^{(g)}(u)\right)\right\| \\
&\qquad - \int_0^1 \left\| H\left(u^* + t\left(u - \mathcal{V}_*^{(g)}(u)\right)\right)\right\| \left\| u - \mathcal{V}_*^{(g)}(u)\right\| dt, \\
&\geq \frac{1}{2}\left\| (I_V - \Pi_g)\, d^{(2)}\Phi(u^*)\left(u - \mathcal{V}_*^{(g)}(u)\right)\left(u - \mathcal{V}_*^{(g)}(u)\right)\right\| - \frac{\epsilon}{2}\left\| u - \mathcal{V}_*^{(g)}(u)\right\|^2. \qquad (3.26)
\end{aligned}$$

From the condition (3.20) we have the following property:

$$\|h\|^2 (I_V - \Pi_g)\, d^{(2)}\Phi(u^*)(h_0)(h_0) \neq 0, \qquad \forall h \in V^{(g)}, h \neq 0,$$

where $h_0 = \frac{1}{\|h\|}h$ is an element of the unity sphere $\mathcal{S}(0_V, 1)$ and the subspace $V^{(g)}$. Since $\mathcal{S}(0_V, 1) \cap V^{(g)}$ is a compact (intersection between the compact $\mathcal{S}(0_V, 1)$ and the closed $V^{(g)}$), then the continuous function $h_0 \mapsto \left\|(I_V - \Pi_g)\, d^{(2)}\Phi(u^*)(h_0)(h_0)\right\|$ attains a lower bound $m > 0$ on $\mathcal{S}(0_V, 1) \cap V^{(g)}$. Consequently, we have:

$$\left\|(I_V - \Pi_g)\, d^{(2)}\Phi(u^*)(h)(h)\right\| \geq m\|h\|^2 \quad \forall h \in V^{(g)}. \qquad (3.27)$$

We know that $u - \mathcal{V}_*^{(g)}(u)$ is in $V^{(g)}$, so we can apply the result of (3.27) with $h = u - \mathcal{V}_*^{(g)}(u)$ and inject it on the lower bound estimate (3.26) with $\epsilon = \frac{m}{2}$:

$$\|(I_V - \Pi_g)(\Phi(u) - u^*)\| \geq \frac{1}{4} m \left\| u - \mathcal{V}_*^{(g)}(u)\right\|^2. \qquad (3.28)$$

Finally, by considering $\alpha = \frac{1}{4}m$, $\beta = \frac{1}{2}\left\|d^{(2)}\Phi(u^*)\right\| + \alpha$ and r the distance associated with $\epsilon = \frac{m}{2}$, we directly obtain the result of Theorem 3.6. □

In the proof of Theorem 3.6, we selected a specific value for the variable ϵ to obtain a pair (α, β) satisfying the inequalities. However, depending on the properties of Φ there might be an optimal choice of ϵ that can keep the size of the domain r sufficiently large while maximizing $\frac{\alpha}{\beta}$, as we will later see the importance of this ratio.

This asymptotic version indicates that, sufficiently close to the solution, the pondering coefficient c defined in Theorem 3.5—which contracts the previous error on the ghost subspace $V^{(g)}$

to bound the current error on the global space—improves as the solution is approached. This improvement reduces any potential gap between the errors on $V^{(g)}$ and $V^{(g)^\perp}$, thereby reinforcing the earlier conclusion on the high accuracy of the values outside the skeleton. In the sequel, we propose a method to recover these values.

We want to see the effect of Theorem 3.6 on the converging sequences generated by the Newton mapping Φ. Hence, the next corollary shows the convergence quotient of a sequence starting in the basin of attraction highlighting the acceleration of the inclination factors squared γ_k^2 that are proven to be strictly bounded from one under the condition (3.20). We also highlight that, while we consider a sequence starting within the basin of attraction, our analysis also applies to any other converging sequence generated by Φ that did not originate in the basin but is examined from the moment it enters it.

COROLLARY 3.7. *Let $\left(u^{(k)}\right)_{k\in\mathbb{N}}$ be a sequence generated by the iterative process Φ that converges to u^*. Then, if the initial guess is close enough to the solution we have:*

$$\left\|u^{(k+1)} - u^*\right\| \leq \beta \left\|u^{(k)} - \mathcal{V}_*^{(g)}(u^{(k)})\right\|^2 = \beta\gamma_k^2 \left\|u^{(k)} - u^*\right\|^2.$$

Where β is the constant defined in Theorem 3.6 and γ_k in Remark 2. If the condition (3.20) holds, then the sequence $(\gamma_k)_{k\in\mathbb{N}_*}$ is bounded above by $\sqrt{1-\left(\frac{\alpha}{\beta}\right)^2}$, with α the constant defined also in Theorem 3.6.

Proof. Let r be the distance defined in Theorem 3.6 and $\left(u^{(k)}\right)_{k\in\mathbb{N}}$ a sequence generated by the iterative process Φ. For an initial guess verifying:

$$\left\|u^{(0)} - \mathcal{V}_*^{(g)}\left(u^{(0)}\right)\right\| < min\left(r, \frac{1}{\beta}\right).$$

With a recursive process, we can prove that $\left\|u^{(k)} - \mathcal{V}_*^{(g)}\left(u^{(k)}\right)\right\|$ is decreasing and thus we can apply the result of Theorem 3.6 for every $u^{(k)}$:

$$\begin{aligned}
\left\|\Phi\left(u^{(k)}\right) - u^*\right\| &\leq \beta \left\|u^{(k)} - \mathcal{V}_*^{(g)}(u^{(k)})\right\|^2, \\
\left\|u^{(k+1)} - u^*\right\| &\leq \beta \left\|u^{(k)} - \mathcal{V}_*^{(g)}(u^{(k)})\right\|^2, \\
\left\|u^{(k+1)} - u^*\right\| &\leq \beta\, \gamma_k^2 \left\|u^{(k)} - u^*\right\|^2.
\end{aligned}$$

And if the condition (3.20) holds, we also have:

$$\left\|(I_V - \Pi_g)\left(u^{(k+1)} - u^*\right)\right\| \geq \alpha \left\|u^{(k)} - \mathcal{V}_*^{(g)}\left(u^{(k)}\right)\right\|^2.$$

Let us now write the expression of γ_{k+1}^2 as follows:

$$\begin{aligned}
\gamma_{k+1}^2 &= \frac{\left\|u^{(k+1)} - \mathcal{V}_*^{(g)}\left(u^{(k+1)}\right)\right\|^2}{\left\|u^{(k+1)} - u^*\right\|^2}, \\
\gamma_{k+1}^2 &= \frac{\left\|u^{(k+1)} - \mathcal{V}_*^{(g)}\left(u^{(k+1)}\right)\right\|^2}{\left\|u^{(k+1)} - \mathcal{V}_*^{(g)}\left(u^{(k+1)}\right)\right\|^2 + \left\|(I_V - \Pi_g)\left(u^{(k+1)} - u^*\right)\right\|^2}.
\end{aligned}$$

The function $x \mapsto \frac{x}{x+\left\|(I_V-\Pi_g)\left(u^{(k+1)}-u^*\right)\right\|^2}$ is increasing in $]0,+\infty[$ and since we know that:

$$\begin{cases} \left\|(I_V-\Pi_g)\left(u^{(k+1)}-u^*\right)\right\|^2 \geq \alpha^2 \left\|u^{(k)}-\mathcal{V}_*^{(g)}(u^{(k)})\right\|^4, \\ \left\|u^{(k+1)}-\mathcal{V}_*^{(g)}\left(u^{(k+1)}\right)\right\|^2 = \gamma_{k+1}^2 \left\|u^{(k+1)}-u^*\right\|^2 \leq \beta^2\gamma_{k+1}^2 \left\|u^{(k)}-\mathcal{V}_*^{(g)}(u^{(k)})\right\|^4. \end{cases}$$

Injecting these two inequalities in the expression of γ_{k+1}^2, we obtain an upper bound on γ_{k+1} in the subsequent manner:

$$\begin{aligned} \gamma_{k+1}^2 &\leq \frac{\beta^2\gamma_{k+1}^2}{\beta^2\gamma_{k+1}^2+\alpha^2}, \\ 1 &\leq \frac{1}{\gamma_{k+1}^2+\left(\frac{\alpha}{\beta}\right)^2}, \quad \text{(Discarding absolute convergence scenario } \gamma_{k+1}=0\text{)} \\ \gamma_{k+1} &\leq \sqrt{1-\left(\frac{\alpha}{\beta}\right)^2}. \end{aligned}$$

Thus, the sequence $(\gamma_i)_{i\in\mathbb{N}_*}$ is bounded above by the term $\sqrt{1-\left(\frac{\alpha}{\beta}\right)^2}$. □

Discussion:

The factor $\frac{\alpha}{\beta}$ gives an idea about the efficiency of the method: the closer to one, the better the rate of convergence is. The latter represents the worst case scenario of the error ratio on $V^{(g)^\perp}$ over the global error on V. Consequently, when the error is mainly located out of the ghost subspace $V^{(g)}$, the convergence gets faster compared to Newton's method applied directly on F. We are aware that the condition (3.20) cannot be checked in practice, but it still provides a theoretical understanding of the factors that influence the acceleration of the process. In particular, the error in the subspace $V^{(g)}$ controls the error in its orthogonal complement. As long as the error remains significant in the latter (α is relatively large), the inclination factors will be smaller, making the acceleration more favorable. In other words, the error restricted to $V^{(g)}$ governs the nonlinearities (the image of $d^{(2)}\Phi\left(u^*\right)$) on both $V^{(g)}$ and $V^{(g)^\perp}$, and when more of it is generated on $V^{(g)^\perp}$ than in $V^{(g)}$, we achieve lower values of $\gamma(u)$.

Now, that we have some insight into how the nonlinear Schwarz method accelerates Newton's method, let us discuss the coefficient β. It is, in fact, related to the operator norm of $d^{(2)}\Phi(u^*)$, which is different from what we would get with the second derivative of the mapping induced by Newton's method applied to the unpreconditioned function F, that we can reference here by β'. Therefore, it is possible that by preconditioning, we end up with $\beta > \beta'$. This may result in some cases, with a well-chosen initial guess, where one converges faster in the unpreconditioned approach than in the preconditioned one. However, since the potential downgrade concerns a constant and the upgrade relies on the factors γ_k that may reach values close to zero, there will always be a starting point zone where RASPEN is better or similar to the standard approach. The bottom line here is that with the results we show, we cannot ensure that the error bound of this preconditioned approach will always be better than the one obtained from the standard approach. We may encounter special cases where the preconditioning negatively affects the nonlinear convergence rate. In practice, we are interested in high-scale discretized PDEs where generally the ghost subspace is of a much lower dimension compared to the global finite element

space. By leveraging the mismatching scales in terms of dimensions, we usually encounter a very small γ_k that mitigates any potential increase in β compared to β'.

3.2. Substructured RASPEN analysis. Let us consider $w = w_\perp + w_g$ an element of U, where $w_\perp$ and w_g are respectively its components on ${V^{(g)}}^\perp$ and $V^{(g)}$. Based on $w_\perp$, we are going to define a domain on which the substructured version of $\mathcal{F}$ will be defined. We shall denote this domain by K and define it as follows:

$$K = \left\{ \quad v_g \in V^{(g)} \quad \middle| \quad w_\perp + v_g \in U \quad \right\} \tag{3.29}$$

Hence, K is an open convex subset of $V^{(g)}$ on which it is possible to differentiate. Thus, the substructured function and its corresponding Jacobian are defined as follows:

$$\begin{aligned} \mathcal{F}^{(g)} : K &\longrightarrow V^{(g)}, & d\mathcal{F}^{(g)}(v_g) : V^{(g)} &\longrightarrow V^{(g)}, \\ v_g &\longmapsto \Pi_g\left(\mathcal{F}\left(w_\perp + v_g\right)\right), & h_g &\longmapsto \Pi_g\left(d\mathcal{F}\left(w_\perp + v_g\right) h_g\right). \end{aligned} \tag{3.30}$$

The function $\mathcal{F}^{(g)}$ is defined on the set K, which is a subset of $V^{(g)}$. Consequently, its elements will be stored in much smaller vectors compared to V. The Jacobian at any element $v_g \in K$ maps from $V^{(g)}$ to $V^{(g)}$, resulting in a matrix whose size corresponds to the dimension of $V^{(g)}$. Despite this, both the substructured function and its Jacobian inherently involve the fine-level function and Jacobian in their formulations. Therefore, the primary advantage lies in the reduced storage requirements for the iterates and the residuals, as well as a smaller Krylov subspace basis to orthogonalize when solving the linear systems. In the remaining of this section, we will introduce some properties of the substructured approach.

Invertibility of the substructured Jacobian:

Newton method applied to the substructured function $\mathcal{F}^{(g)}$ will always be well-posed as long as it is well-posed for $\mathcal{F}$, i.e., $d\mathcal{F}(v)$ is invertible for any element $v \in U$. We shall prove that $d\mathcal{F}^{(g)}(v_g)$ is also invertible for any $v_g \in K$:

PROPOSITION 3.8. *Assuming that for any $v \in U$, $d\mathcal{F}(v)$ is invertible. Then, $d\mathcal{F}^{(g)}$ is also invertible on K that is:*

$$Ker\left(d\mathcal{F}^{(g)}(v_g)\right) = \{0_V\}, \quad \forall v_g \in K. \tag{3.31}$$

Proof. Let (h_g, v_g) in $\left(V^{(g)}, K\right)$ and $v = w_\perp + v_g$, we have:

$$\begin{aligned}
d\mathcal{F}^{(g)}(v_g)\, h_g = 0_V &\Longrightarrow \Pi_g\left(d\mathcal{F}(v)\, h_g\right) = 0_V, \quad \left(\text{By definition of } d\mathcal{F}^{(g)}(v_g)\right) \\
&\Longrightarrow \begin{cases} \Pi_g\left(d\mathcal{F}(v)\, h_g\right) = 0_V, \\ d\mathcal{F}(v)^2 h_g = d\mathcal{F}(v)\, h_g, \end{cases} \quad \text{(Using Lemma 3.1 with } u = d\mathcal{F}(v)\, h_g) \\
&\Longrightarrow \begin{cases} \Pi_g\left(d\mathcal{F}(v)\, h_g\right) = 0_V, \\ d\mathcal{F}(v)\, h_g = h_g, \end{cases} \\
&\Longrightarrow \Pi_g(h_g) = 0_V, \\
&\Longrightarrow h_g = 0_V, \quad \left(\text{Since } h_g \in V^{(g)}\right). \square
\end{aligned}$$

Thus, $d\mathcal{F}^{(g)}(v_g)$ is invertible, allowing us to safely define the substructured Newton mapping $\Phi^{(g)}$, which is defined for every element $v_g \in K$, as follows:

$$\Phi^{(g)}(v_g) = v_g - \left(d\mathcal{F}^{(g)}(v_g)\right)^{-1} \mathcal{F}^{(g)}(v_g).$$

Equivalence between Newton and substructured Newton:

Next, we will show the following relationship between the RASPEN mapping Φ and the SRASPEN mapping $\Phi^{(g)}$:

PROPOSITION 3.9. *Let $v_g \in K$, we have:*

$$\Phi^{(g)}(v_g) = \Pi_g\left(\Phi(w_\perp + v_g)\right) \tag{3.32}$$

Proof. Let $v_g \in K$ and $v = w_\perp + v_g$. In [4] this property was proven for a prolongation to the volume with zero i.e. $w_\perp = 0$. We shall recall the proof for any $w_\perp \in V^{(g)^\perp}$ starting from the definition of $\Phi(v)$:

$$\begin{aligned}
\Phi(v) &= v - \left(d\mathcal{F}(v)\right)^{-1} \mathcal{F}(v), \\
d\mathcal{F}(v)\left(v - \Phi(v)\right) &= \mathcal{F}(v).
\end{aligned}$$

We decompose $v - \Phi(v)$ into two orthogonal components using the projection operator Π_g:

$$\begin{aligned}
&d\mathcal{F}(v)\left(\Pi_g\left(v - \Phi(v)\right) + \left(I_V - \Pi_g\right)\left(v - \Phi(v)\right)\right) = \mathcal{F}(v), \\
&d\mathcal{F}(v)\,\Pi_g\left(v - \Phi(v)\right) + d\mathcal{F}(v)\left(I_V - \Pi_g\right)\left(v - \Phi(v)\right) = \mathcal{F}(v).
\end{aligned}$$

From Lemma 3.1, we have $d\mathcal{F}(v)(I_V - \Pi_g) = (I_V - \Pi_g)$ leading to:

$$d\mathcal{F}(v)\,\Pi_g(v - \Phi(v)) + (I_V - \Pi_g)(v - \Phi(v)) = \mathcal{F}(v).$$

We apply Π_g on both sides and because $\Pi_g(I_V - \Pi_g) = 0_{V\to V}$, we get:

$$\begin{aligned}
\Pi_g(d\mathcal{F}(v)\,\Pi_g(v - \Phi(v))) &= \Pi_g(\mathcal{F}(v)),\\
d\mathcal{F}^{(g)}(v_g)\,[\Pi_g(v) - \Pi_g(\Phi(v))] &= \mathcal{F}^{(g)}(v_g),\\
v_g - d\mathcal{F}^{(g)}(v_g)^{-1}\,\mathcal{F}^{(g)}(v_g) &= \Pi_g(\Phi(v)), \quad (\ \Pi_g(v) = v_g\ ),\\
\Phi^{(g)}(v_g) &= \Pi_g(\Phi(w_\perp + v_g)).
\end{aligned}$$

This concludes the proof. □

In terms of converging sequences generated by the mappings Φ and $\Phi^{(g)}$, this property ensures that their corresponding iterates have the same values on the subspace $V^{(g)}$, that is for any sequence $\left(u^{(k)}\right)_{k\in\mathbb{N}}$ generated by Φ, the sequence $\left(u_g^{(k)}\right)_{k\in\mathbb{N}}$ generated by $\Phi^{(g)}$ with the initial guess $u_g^{(0)}$ such that $u_g^{(0)} = \Pi_g\left(u^{(0)}\right)$, verifies $u_g^{(k)} = \Pi_g\left(u^{(k)}\right)$ for any $k \in \mathbb{N}$. We can obtain this result recursively as follows:

$$\begin{aligned}
u_g^{(k)} = \Pi_g\left(u^{(k)}\right) &\Longrightarrow \Pi_g\left(w_\perp + u_g^{(k)}\right) = \Pi_g\left(u^{(k)}\right),\\
&\Longrightarrow \Phi\left(w_\perp + u_g^{(k)}\right) = \Phi\left(u^{(k)}\right), \quad (\text{ Using Theorem 3.4 })\\
&\Longrightarrow \Pi_g\left(\Phi\left(w_\perp + u_g^{(k)}\right)\right) = \Pi_g\left(u^{(k+1)}\right),\\
&\Longrightarrow \Phi^{(g)}\left(u_g^{(k)}\right) = \Pi_g\left(u^{(k+1)}\right), \quad (\text{Using Equation (3.32)})\\
&\Longrightarrow u_g^{(k+1)} = \Pi_g\left(u^{(k+1)}\right). \qquad (3.33)
\end{aligned}$$

Consequently, we can now work only on the function $\mathcal{F}^{(g)}$ to obtain an approximation on $V^{(g)}$. Then, when the latter becomes sufficiently accurate, obtain a global approximation through nonlinear local solves.

Choice of $w_\perp$ for a better performance:

Until now, the choice of $w_\perp$ did not matter in the proof of equivalence between Newton and substructured Newton. From a theoretical point of view, it is the case, but when it comes to practise, the choice of $w_\perp$ is crucial. Indeed, for a chosen nonlinear solver of the nonlinear local problems, the quality of the local initial guess depends on $w_\perp$, which makes it control the number of iterations needed to obtain the local corrections or even ensure the convergence of the nonlinear local solver. Hence, we need to choose $w_\perp$ as accurate as possible. Since the iterates values of the substructured sequence are the same for any $w_\perp$, we can choose a different $w_\perp$ on each iteration that we denote by $w_\perp^{(k)}$ and we will denote the full iterates associated with the substructured Newton as:

$$v^{(k)} = w_\perp^{(k)} + u_g^{(k)}. \qquad (3.34)$$

For any $k \in \mathbb{N}_*$, the value of $w_\perp^{(k)}$ will be determined by one of the subsequent choices:

- **Strategy 1: First initial guess initialization (used in [4]).**

$$w_{\perp}^{(k)} = (I_V - \Pi_g)\, u^{(0)}.$$

- **Strategy 2: Recovering the action of the nonlinear preconditioner (suggested in [4]).**

$$w_{\perp}^{(k)} = (I_V - \Pi_g) \left(v^{(k-1)} - \mathcal{F}\left(v^{(k-1)} \right) \right),$$

  where $v^{(k)}$ is defined in (3.34).

- **Strategy 3: Recovering the Newton iterate (our proposition).**

$$w_{\perp}^{(k)} = (I_V - \Pi_g) \left(v^{(k-1)} - \mathcal{F}\left(v^{(k-1)} \right) - d\mathcal{F}\left(v^{(k-1)} \right) d_g^{(k)} \right), \tag{3.35}$$

  where $d_g^{(k)} = u_g^{(k)} - u_g^{(k-1)}$ is the substructured Newton update, corresponding to the values update obtained on the skeleton.

The first strategy follows the method used in [4], where the component of the solution on $V^{(g)^{\perp}}$ is set to the initial guess $u^{(0)}$ (which is usually zero when no prior approximation of the solution is available at the start), thus, stays constant throughout the iterations. Although this approach is straightforward to implement, it carries the risk of not enabling the convergence of some nonlinear local problems. Even when convergence is achieved, many nonlinear local iterations may be required to reach the solution. This issue can persist throughout all the nonlinear global Newton iterations, as no significant improvement is observed as the solution is approached.

The second strategy also mentioned in [4] involves initializing the local solvers with the action of the nonlinear preconditioner from the previous step, which is available through the function image of the previous iterate. By using this approach, we can improve the performance of the nonlinear local solver, as the initial guess becomes more accurate with each outer Newton iteration, reducing the number of iterations that the nonlinear local solver requires. However, this method is still not as effective as the approximation obtained when solving with the full Newton mapping, *i.e.*, RASPEN, because the approximation provided by the nonlinear preconditioner accelerated by Newton's method is generally more accurate than its non-accelerated version, especially when Newton's method reaches quadratic convergence.

We propose a third strategy, which allows us to recover the RASPEN iterates on the entire domain so that the values associated with interior domain unknowns can be used to define the initial guesses for the nonlinear local subdomain solvers in the next SRASPEN iteration. As we can see in Equation (3.35), at the end of iteration $k-1$, $d_g^{(k)}$ is available. Since $\mathcal{F}^{(g)}\left(u_g^{(k-1)}\right)$ and $d\mathcal{F}^{(g)}\left(u_g^{(k-1)}\right)$ have been computed during the $k-1$ iteration, $\mathcal{F}\left(v^{(k-1)}\right)$ and $d\mathcal{F}\left(v^{(k-1)}\right)$ are also accessible by definition of $\mathcal{F}^{(g)}$ and $d\mathcal{F}^{(g)}$. Therefore, this update is feasible once the $(k-1)$-th iteration is complete, allowing us to define a very good initial guess for the nonlinear local solves of the k-th iteration. We next prove that the third choice enables us to recover the full iterates of the fine-level Newton:

PROPOSITION 3.10. *Let* $\left(u^{(k)}\right)_{k\in\mathbb{N}}$ *and* $\left(u_g^{(k)}\right)_{k\in\mathbb{N}}$ *be two sequences generated, respectively by the RASPEN mapping* Φ *and the SRASPEN mapping* $\Phi^{(g)}$*, such that the following initial condition is satisfied:*

$$v^{(0)} = w_\perp^{(0)} + u_g^{(0)} = u^{(0)}. \tag{3.36}$$

Then, with the definition (3.35) of $w^{(k)}$*, the sequence* $v^{(k)} = w_\perp^{(k)} + u_g^{(k)}$ *is equivalent to* $u^{(k)}$*, meaning that:*

$$v^{(k)} = u^{(k)}, \quad \forall k \in \mathbb{N}. \tag{3.37}$$

Proof. We will demonstrate recursively that $v^{(k)} = u^{(k)}$ for any $k \in \mathbb{N}$. By the assumption (3.36), we have $v^{(0)} = u^{(0)}$. Let $k \in \mathbb{N}_*$, we shall assume that $v^{(k-1)} = u^{(k-1)}$ and prove $v^{(k)} = u^{(k)}$. We know from the result (3.33) that for any $j \in \mathbb{N}, u_g^{(j)} = \Pi_g\left(u^{(j)}\right)$. Thus, if the iterates have this property, their corresponding updates $\left(d^{(j)}\right)_{j\in\mathbb{N}_*}$ will maintain it as well and particularly at the k-th iteration:

$$\begin{aligned} d_g^{(k)} = \Pi_g\left(d^{(k)}\right) &\implies d^{(k)} = d_g^{(k)} + d_\perp^{(k)}, \quad \text{with } d_\perp^{(k)} = (I - \Pi_g)\left(d^{(k)}\right), \\ &\implies d\mathcal{F}\left(u^{(k-1)}\right) d^{(k)} = d\mathcal{F}\left(u^{(k-1)}\right) d_g^{(k)} + d\mathcal{F}\left(u^{(k-1)}\right) d_\perp^{(k)}. \end{aligned}$$

We know that the Newton update $d^{(k)}$ verifies $d\mathcal{F}\left(u^{(k-1)}\right) d^{(k)} = -\mathcal{F}\left(u^{(k-1)}\right)$, and also from Lemma 3.1, we have $d\mathcal{F}(u^{(k-1)})\, d_\perp^{(k)} = d_\perp^{(k)}$. Consequently:

$$\begin{aligned} &-\mathcal{F}\left(u^{(k-1)}\right) = d\mathcal{F}\left(u^{(k-1)}\right) d_g^{(k)} + d_\perp^{(k)}, \\ &d_\perp^{(k)} = (I - \Pi_g)\left(-\mathcal{F}\left(u^{(k-1)}\right) - d\mathcal{F}\left(u^{(k-1)}\right) d_g^{(k)}\right). \end{aligned}$$

Now, we use the expression of $w_\perp^{(k)}$ in (3.35) where $v^{(k-1)}$ is replaced by $u^{(k-1)}$, which leads to:

$$\begin{aligned} d_\perp^{(k)} &= w_\perp^{(k)} - (I - \Pi_g)\left(u^{(k-1)}\right), \\ w_\perp^{(k)} &= (I - \Pi_g)\left(d^{(k)} + u^{(k-1)}\right), \\ w_\perp^{(k)} &= (I - \Pi_g)\left(u^{(k)}\right), \\ v^{(k)} &= (I - \Pi_g)\left(u^{(k)}\right) + u_g^{(k)}, \\ v^{(k)} &= u^{(k)}. \end{aligned}$$

This concludes the proof of the equivalence between the full iterates of RASPEN and SRASPEN with an initial guess of the third strategy. □

With this option, we will be able to initialize the nonlinear local problems accurately, as in RASPEN, while also benefiting from solving a smaller linear system on the skeleton instead of the volume. Additionally, we highlight that the formulation of the third option is very efficient, requiring minimal extra computational cost and no additional memory storage, as the values will be distributed and stored directly in the local vectors corresponding to the solutions of the subdomains nonlinear problems. Below, an algorithm for the third strategy is provided:

Algorithm 1 SRASPEN

$v^{(0)} = u^{(0)}$
$u_g^{(0)} = \Pi_g \left(u^{(0)}\right)$
for $k = 0$ **until convergence do**
  Compute $\mathcal{F}\left(v^{(k)}\right)$
  Deduce $\mathcal{F}^{(g)}\left(u_g^{(k)}\right)$ as $\Pi_g \left(\mathcal{F}\left(v^{(k)}\right)\right)$
  Compute action of $d\mathcal{F}\left(v^{(k)}\right)$
  Deduce action of $d\mathcal{F}^{(g)}\left(u_g^{(k)}\right)$ as $d\mathcal{F}^{(g)}\left(u_g^{(k)}\right) h_g = \Pi_g \left(d\mathcal{F}\left(v^{(k)}\right) h_g\right), \quad \forall h_g \in V^{(g)}$
  Solve $d\mathcal{F}^{(g)}\left(u_g^{(k)}\right) d_g^{(k)} = -\mathcal{F}^{(g)}\left(u_g^{(k)}\right)$, a linear system on $V^{(g)}$ (on the skeleton)
  Update $u_g^{(k+1)} \leftarrow u_g^{(k)} + d_g^{(k)}$
  Update $v^{(k+1)} \leftarrow v^{(k)} + d_g^{(k)} - \mathcal{F}\left(v^{(k)}\right) - d\mathcal{F}\left(v^{(k)}\right) d_g^{(k)}$
end for

4. Numerical Illustrations.

4.1. Test Models. To evaluate the performance of the numerical convergence properties of RASPEN and SRASPEN and for the sake of comparison with already published results on these methods we consider two academic problems. The first problem, also considered in [4, 11], is a 1D Forchheimer porous media problem. The second model problem, borrowed from [4], is a 2D nonlinear diffusion problem. All convergence histories presented in the numerical experiments correspond to the relative error of the preconditioned function on the y-axis, defined at the k-th step as:

$$\text{Relative error} = \frac{\left\|\mathcal{F}\left(u^{(k)}\right)\right\|}{\left\|\mathcal{F}\left(u^{(0)}\right)\right\|}, \tag{4.1}$$

where $u^{(k)}$ represents the iterate at the k-th step. The threshold for the stopping criterion on the relative error defined by Equation (4.1) is set to 10^{-10} for the Forchheimer example. In order to illustrate the nonlinear slow pre-asymptotic phase of classical Newton (which, in our case, will manifest in the nonlinear local solvers), we choose an initial guess with a large initial norm, *i.e.*, $\|\mathcal{F}(u^{(0)})\|$, and consequently set a much smaller threshold of 10^{-20} for the nonlinear diffusion problem.

Since we intend to illustrate the impact of the initial guess chosen for the solution of the nonlinear local problems, we define a different stopping criterion for the nonlinear local solvers that is independent from the initial guess. Consequently the stopping criterion for the nonlinear local solver is defined by a threshold on the absolute error associated with the local approximation functions $F^{(i)}_{u^{(k)}}$ when solving the subproblems (2.5). This threshold is set to 10^{-12} for the Forchheimer case and 10^{-16} for the nonlinear diffusion problem. On the latter example, during the early stages when the initial guess is still far from the solution, the nonlinear local solver may stagnate. In such cases, convergence is determined based on a small solution step criterion. Finally, the global linear system to solve at each step is solved by GMRES [30] where the relative stopping criteria is 10^{-8} for the Forchheimer problem and 10^{-12} for the nonlinear diffusion problem.

The code used in this case is implemented in parallel using Firedrake [19] for mesh generation and finite element discretization of the PDEs. The solver runs on a parallel distributed MPI architecture through petsc4py [7], which provides Python bindings for the PETSc toolkit [1].

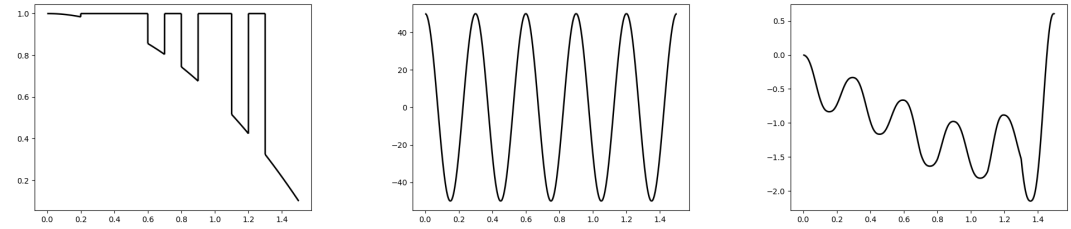
In all the parallel experiments, each subdomain is assigned to one MPI process, with no multi-threading at the subdomain level. At each iteration of the nonlinear subdomain problem, the corresponding linear system is solved using a direct sparse LU factorization, which is computationally affordable at the subdomain level. These subdomain factorizations are stored and reused for the subdomain linear solves involved in the GMRES method through the action of $d\mathcal{F}$. The computations were performed on the Kraken cluster at CERFACS, which comprises 4,284 cores distributed across 119 compute nodes, each equipped with two 18-core Intel Xeon Gold 6140 processors running at 2.3 GHz.

4.1.1. Forchheimer. In this section, we consider the Forchheimer model [6, 17, 32], which generalizes Darcy's Law by adding a deviation term to account for inertial effects. Following the approach in [4, 11], this model serves as a reference to first demonstrate how the RASPEN method accelerates nonlinear convergence compared to classical Newton, and to discuss various convergence indicators in terms of their cost and scalability with respect to the number of subdomains.

The Forchheimer model in the interval $\Omega = [0, L]$ is expressed as:

$$\begin{cases} \left(q\left(-\lambda\left(x\right)u\left(x\right)\right)\right)' = f\left(x\right) & \text{in} \quad \Omega, \\ u(0) = u_0, \\ u(L) = u_L, \end{cases}$$

where $q\left(v\right) = \operatorname{sgn}\left(v\right)\frac{-1+\sqrt{1+4\rho|v|}}{2\rho}$, λ is the permeability of the medium, and ρ is the Forchheimer coefficient controlling the nonlinear aspect of the equation. When $\rho \to 0^+$, Darcy's Law is recovered. The model will be discretized using a first-order finite element method. We will consider the two setups defined in [11] as illustrated in Figure 4.1 and Figure 4.2.



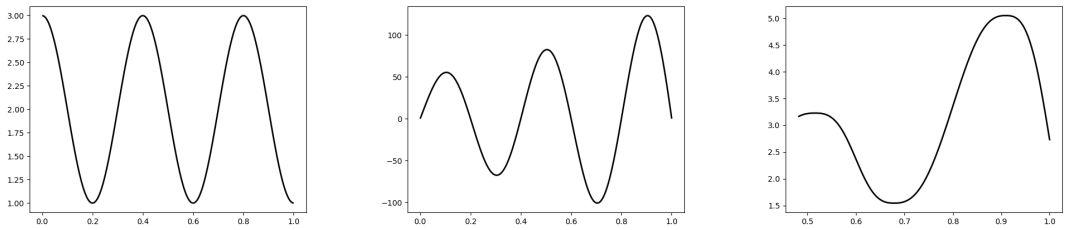
(a) Permeability Function $\lambda(x)$ (b) Source Term $f(x)$ (c) Solution $u(x)$

Fig. 4.1: Forchheimer 1 - permeability function, source term, and solution.

4.1.2. Nonlinear Diffusion. Our 2D test model is a nonlinear diffusion problem with parameters and initial conditions similar to those studied in [4]. It is defined in the unit square domain $\Omega = (0, 1)^2$ and writes:

$$\begin{cases} -\nabla \cdot \left[\left(1 + u(x)^2\right)\nabla u(x)\right] = f(x), & \text{in } \Omega, \\ u(x) = g(x), & \text{on } \partial\Omega, \end{cases}$$

where f and g are, respectively, the source term and the boundary condition chosen such that the exact solution is $u(x) = \sin(\pi x)\sin(\pi y)$. The initial guess is set to $u_0(x) = 10^5$, placing the



(a) Permeability Function $\lambda(x)$ (b) Source Term $f(x)$ (c) Solution $u(x)$

Fig. 4.2: Forchheimer 2 - permeability function, source term, and solution.

starting point far from the solution to illustrate the slow convergence rate in the pre-asymptotic phase of the standard Newton's method.

4.2. Numerical performances. In all the numerical experiments below, we will use the following notations to denote different performance indicators:

$\mathbf{N}$	Number of subdomains.
$\mathbf{k}$	Number of nonlinear RASPEN iterations.
$\mathbf{k}_{\text{New}}$	Number of nonlinear classical Newton iterations.
$\mathbf{k}_C$	Cumulative number of local Newton iterations for computing the local corrections C_i. At each outer step, we add the maximum number of nonlinear iterations across the subdomains, accounting for the nonlinear subdomain solution unbalance.
$\mathbf{k}_G$	Cumulative number of GMRES iterations to solve the global linear system associated with RASPEN.
$\mathbf{k}_{\text{tot}}$	Cumulative number of linear local solves in a parallel setting: $\mathbf{k}_{\text{tot}} = \mathbf{k}_G + \mathbf{k}_C$. Here, $\mathbf{k}_C$ contributes because each nonlinear local solve includes one linear local direct solve.

Table 4.1: Notations used to evaluate the performance of the different algorithms.

For the sake of comparison, we will report on these indicators for both methods, RASPEN and SRASPEN in the next sections.

4.2.1. The performance of RASPEN. In this section, we analyze the results of RASPEN applied to the Forchheimer problem by discussing each of the cost indicators defined in Table 4.1. We first investigate the impact of the size of the overlap on the performance of the RASPEN method in a weak scalability setting. The results are reported in Table 4.2 where it can be observed that increasing the overlap slightly reduces the number of GMRES iterations while increasing the number of nonlinear local iterations $\mathbf{k}_C$, since the subdomains become larger. The impact of the overlap on the total number of nonlinear iterations varies. Increasing the overlap moves the ghost nodes, thereby changing the ghost subspace, which can result in either a larger or smaller initial residual in a in-predictable fashion. Similarly, in a weak scalability approach, adding more subdomains changes the location of the ghost nodes, leading to fluctuations in the number of RASPEN nonlinear iterations $\mathbf{k}$. However, this number does not grow as significantly

$\boldsymbol{\xi = h_N}$						$\boldsymbol{\xi = 4\ h_N}$					
N	**k**	$\mathbf{k}_{New}$	$\mathbf{k}_C$	$\mathbf{k}_G$	$\mathbf{k}_{tot}$	**N**	**k**	$\mathbf{k}_{New}$	$\mathbf{k}_C$	$\mathbf{k}_G$	$\mathbf{k}_{tot}$
2	4	18	29	12	41	2	3	18	26	9	35
4	4	25	33	28	61	4	4	25	31	28	59
8	5	34	36	80	116	8	5	34	37	79	116
16	8	47	75	264	339	16	7	47	72	224	296
32	9	65	128	606	734	32	7	65	134	460	594
64	11	91	124	1496	1620	64	9	91	107	1188	1295
128	7	127	48	1855	1903	128	7	127	49	1768	1817
$\boldsymbol{\xi = 6\ h_N}$						$\boldsymbol{\xi = 10\ h_N}$					
N	**k**	$\mathbf{k}_{New}$	$\mathbf{k}_C$	$\mathbf{k}_G$	$\mathbf{k}_{tot}$	**N**	**k**	$\mathbf{k}_{New}$	$\mathbf{k}_C$	$\mathbf{k}_G$	$\mathbf{k}_{tot}$
2	3	18	26	9	35	2	3	18	28	9	37
4	4	25	32	28	60	4	4	25	34	28	62
8	5	34	37	75	112	8	5	34	39	75	114
16	8	47	85	256	341	16	9	47	104	288	392
32	9	65	137	591	728	32	10	65	159	639	798
64	9	91	108	1169	1277	64	10	91	125	1238	1363
128	9	127	50	1709	1759	128	7	127	51	1604	1655

Table 4.2: RASPEN results for the conditions Forchheimer 1 of Figure 4.1 under a weak scalability approach of 100 degrees of freedom per subdomain and different sizes of overlap ξ in terms of the mesh size h_N that is dependent on the number of subdomain due to weak scalability.

as the number of nonlinear exact Newton iterations, $\mathbf{k}_{New}$. The number of nonlinear local iterations, $\mathbf{k}_C$, depends strongly on the nonlinear rate of convergence: the faster the iterates converge to the solution, the fewer nonlinear local iterations are needed, as the nonlinear local iterations on the subdomains start from a good local initial guess.

The classical slowdown of information transfer between subdomains as N increases is also present, as shown by the increasing number of GMRES iterations with respect to N. Both $\mathbf{k}_C$ and $\mathbf{k}_G$ represent the local solves, and their sum gives the total number of local solutions performed throughout the entire RASPEN process. Since a direct solver is chosen for the local linear solves of the nonlinear subdomains problems, $\mathbf{k}_C$ counts for the number of factorizations and the forward/backward substitutions, with the factorization phase being the most costly. On the other hand, $\mathbf{k}_G$ counts only for the forward/backward substitutions since, in the linear system, the local parts of the Jacobian do not change and need to be factorized only once at the first GMRES iteration. Consequently, to compare the Newton-Krylov method preconditioned with an additive Schwarz approach to RASPEN in terms of the number of factorizations, we need to compare $\mathbf{k}_{en}$ with $\mathbf{k} + \mathbf{k}_C$. However, the number of full non-restarted GMRES iterations incurs additional

costs due to the quadratic increase of the cost of the modified Gram-Schmidt orthogonalisation process, which affects both the amount of computations and of communications. Unlike the cost of factorizations, which remains constant in weak scalability and decreases in strong scalability, the cost of communications increases with the number of processors. Thus, depending on the number of subdomains and their size, one cost indicator may be more or less significant than the other to provide insight on the parallel time to solution.

4.2.2. The performance of SRASPEN. In this section, we present SRASPEN numerical experiments to illustrate several key points about the initialization strategies for the nonlinear local solvers. First, we demonstrate the insensitivity of SRASPEN's nonlinear global convergence to the choice of initialization strategy due to our choice of the local stopping criterion. Second, we highlight how these strategies impact the convergence rate of the nonlinear local solves. Third, we analyze the linear convergence behavior of GMRES, showing that the number of iterations remains independent of the initial guess of the nonlinear local solves. Fourth, we investigate the effect of increasing the number of subdomains on the global convergence and the nonlinear local convergence for each strategy. Fifth, we report on the execution times for the different strategies. Finally, we compare our version of SRASPEN with standard RASPEN, demonstrating that SRASPEN provides equivalent nonlinear global convergence while providing a more effective solution technique thanks to the global memory savings and computational efficiency of the GMRES iterations that are performed on a lower dimension problem. **Impact of the initial guess in the nonlinear local solves on the overall global nonlinear convergence:** a preliminary interpretation of Figure 4.3 suggests the equivalence of the nonlinear global convergence, as shown in the left graph. The dots (corresponding to the outer nonlinear iterations) on each curve (Strategy 1, 2 and 3) align approximately along the same horizontal lines, indicating that they achieve similar accuracy at each step. However, it can be observed in the right graph that the dots do not strictly correspond to the same relative error (they are horizontally slightly misaligned). This difference arises because the nonlinear local solutions vary slightly; the initial guesses differ, and although the convergence criterion is absolute (meaning it does not explicitly depend on the initial guess/error), we cannot guarantee that the computed solution will be identical. In any case, the local initial guess does not significantly affect the global convergence

Impact of the nonlinear local initial guess on the nonlinear local convergence: in the graphs of Figure 4.3 showing the nonlinear local iterations for each strategy, the blue curve corresponding to the first strategy shows a significantly higher number of iterations compared to the orange and green curves. The orange curve corresponds to the second strategy, where the action of the nonlinear restricted Schwarz preconditioner from the previous step is reused as an initial guess in the next iteration. This strategy results in fewer total nonlinear local iterations compared to the first one, because the initial guess becomes more accurate through the nonlinear global iterations as it gets closer to the solution. This is reflected in the shape of the curve, which becomes vertical toward the end of convergence, where the value of $\mathbf{k}_C$ barely increases. Finally, the green curve, corresponding to the third strategy, outperforms the other two, as it leverages the recovery of the full iterates of RASPEN, which are a Newton-accelerated version of the second strategy iterates. Consequently, the total number of nonlinear local iterations decreases sharply and stagnates earlier compared to the orange curve.

Impact of the nonlinear local initial guess on the GMRES convergence: Figure 4.4 shows the number of GMRES iterations for each strategy, which should remain unaffected by changes in the initial guess. This is confirmed, as the points on each curve align closely along the same vertical lines. However, we still observe differences in the relative error at each step, similar to those in Figure 4.3, due to the same numerical reasons previously mentioned for the impact on the global nonlinear convergence.

While increasing the number of subdomains: in Figure 4.5 and as shown before, the third strategy initial guess achieves significant savings, maintaining an approximate ratio of 2.20 compared to the second strategy and around 8.10 compared to the first strategy. Most importantly, this saving ratio remains consistent as the number of subdomains increases, which leads to a similar reduction on the cumulative time of the nonlinear local solves throughout convergence. This is because the cost of each nonlinear local iteration is equivalent to that of a direct linear local solve of the same local size in a weak scalability approach. In this case, the first strategy highlights the problem of continuing to initialize with the same poor initial guess which is in that case $u_0 = 10^5$. The second strategy improves the nonlinear local convergence but still can not recover directly the first RASPEN iterate which is very powerful while the third strategy does.

Execution time for the different local initial guess strategies: Table 4.3 presents the execution time for the Forchheimer problem under each initial guess strategy as the local size $\boldsymbol{n}_{\text{loc}}$ increases. As anticipated from the previous figures, the execution time reflects the savings achieved in the nonlinear local iterations. Using the third strategy for the initial guess results in the best time to solution performance. More importantly, in terms of robusteness, the table shows that when the number of degrees of freedom per subdomain exceeds 200, both the first and the second strategies completely lose local and, consequently, global convergence. Therefore, when the nonlinear convergence rate degrades at high-scale discretizations, as in this case, having a robust initial guess strategy is crucial to ensure convergence of the nonlinear solver. In Table 4.4, we compare the execution times of SRASPEN in the 2D nonlinear diffusion problem for each type of initial guess. As expected, third strategy demonstrates up to a 6.16x speed-up compared to the first strategy and a 1.88x improvement over the second strategy. Since the local linear solve associated with each nonlinear local iteration involves not only forward/backward substitutions but also factorization for each linear local solve, $\mathbf{k}_C$ is the dominant component in terms of cost compared to the linear local solves in $\mathbf{k}_G$. This makes the reduction in the number of nonlinear local solves strongly influence the overall execution time. Nevertheless, the speed-up factor decreases as the number of subdomains increases. This decline is primarily due to the GMRES solver requiring more iterations, which makes its execution time the dominant component of the overall runtime, overshadowing the impact on the execution time of nonlinear local solves.

Execution time comparison between RASPEN and SRASPEN: Now that we have established that the SRASPEN version with the third strategy for the initial guess is the most effective, we will compare it with the standard RASPEN in terms of global nonlinear convergence and execution time. Figure 4.6 illustrates the equivalence of the full iterates between RASPEN and SRASPEN, as demonstrated in Section 3.2. However, while the full iterates are similar, Table 4.5 shows that SRASPEN significantly outperforms RASPEN in execution time, especially as the number of subdomains increases. This performance gain is largely attributed to the fact that SRASPEN operates in a lower-dimensional space. In SRASPEN, the orthogonalization is performed on a skeleton-sized Krylov basis rather than a volume-sized one, resulting in faster execution. Furthermore, SRASPEN benefits from the robust initial guess provided by RASPEN, which reduces the time required for nonlinear local solves. Using the same initial guess for both RASPEN and SRASPEN—thus incurring the same cost for computing the subdomain correction—further highlights the advantages of substructuring, particularly during the GMRES phase.

5. Conclusion. Throughout the paper, the convergence of the RASPEN method was analyzed, emphasizing the impact of nonlinear domain decomposition on the convergence rate in both the pre-asymptotic and asymptotic phases. This convergence study revealed that a single step is sufficient for the volume error to become smaller than the previous error on the skeleton.

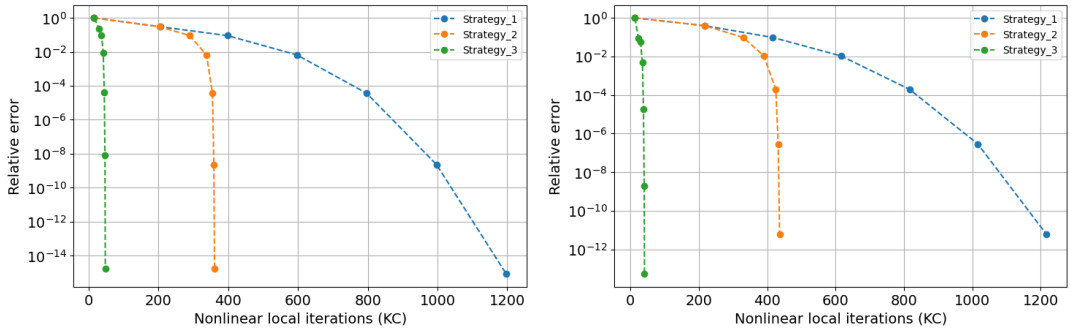


Fig. 4.3: Comparing the nonlinear local iterations $\mathbf{k}_C$ of SRASPEN with the different initial guess strategies for the Forchheimer 2 conditions shown in Figure 4.2. The graph on the left corresponds to 20 subdomains while the one on the right corresponds to 50 subdomains. The overlap is $8h$ and the mesh size is $h = 10^{-3}$.

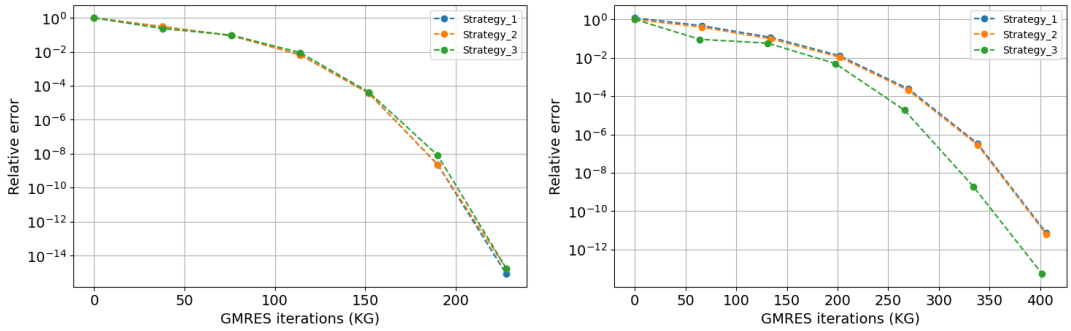


Fig. 4.4: Comparing the number of GMRES iterations ($\mathbf{k}_G$) of SRASPEN with different initial guess strategies for the Forchheimer 2 conditions shown in Figure 4.2. The left graph corresponds to 20 subdomains, while the right graph corresponds to 50 subdomains. The overlap is set to $8h$, and the mesh size is $h = 10^{-3}$.

Initial Guess	**Strategy 1**		**Strategy 2**		**Strategy 3**	
	$\mathbf{n_{loc}}$	**Time(s)**	$\mathbf{n_{loc}}$	**Time(s)**	$\mathbf{n_{loc}}$	**Time(s)**
	25	0.89	25	0.47	25	0.21
	50	1.12	50	0.69	50	0.31
	100	2.14	100	1.67	100	0.41
	200	∞	200	∞	200	0.64

Table 4.3: Parallel execution time comparison between initial guess strategies for the Forchheimer 2 conditions of Figure 4.2 with 16 subdomains and an increasing subdomain size $\mathbf{n_{loc}}$.

This observation led to the conclusion that the values in the rest of the volume remain as accurate as those on the skeleton.

In the case of SRASPEN, which operates only on the skeleton, these values are not computed.

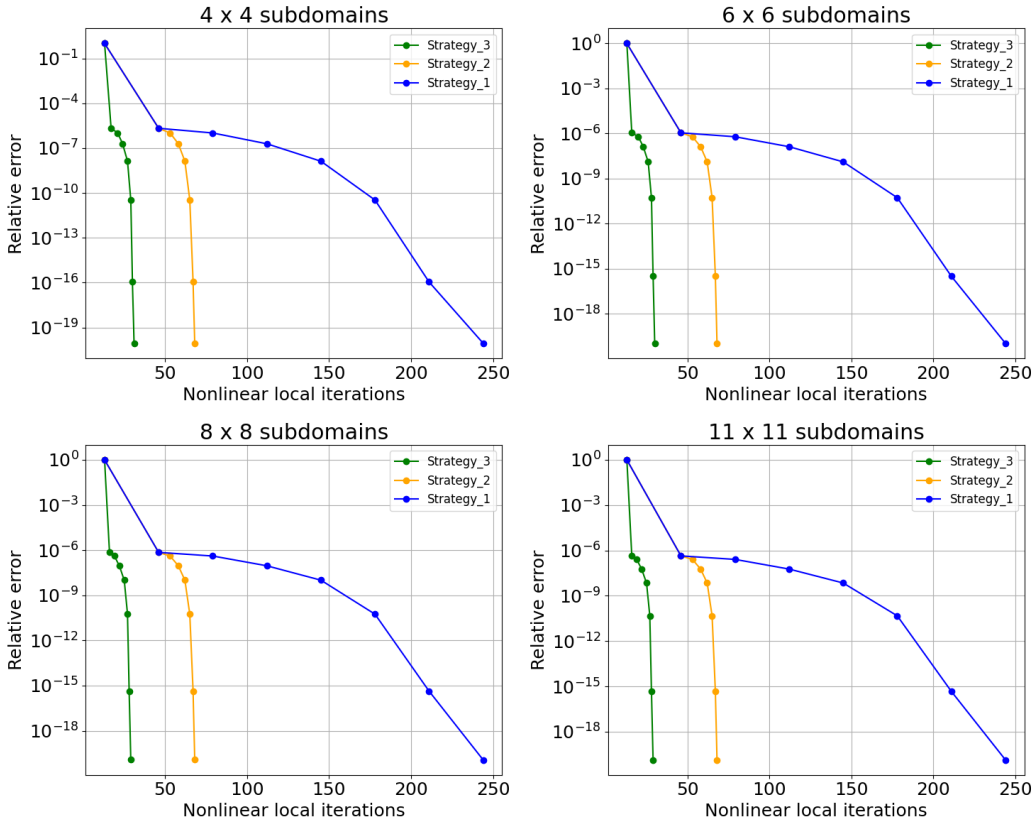


Fig. 4.5: Comparing the number of nonlinear local iterations ($\mathbf{k}_C$) of SRASPEN with different choices for the initial guess in the 2D nonlinear diffusion case with 4×4, 6×6, 8×8 and 11×11 subdomains. The overlap is $8h$ and each subdomain has 9.10^4 degrees of freedom.

Subdomains	**4 × 4**	**6 × 6**	**8 × 8**	**11 × 11**
Strategy 1	411.83 s (6.16)	434.70 s (5.70)	515.43 s (5.64)	**643.56 s** (4.41)
Strategy 2	125.41 s (1.88)	143.17 s (1.88)	160.25 s (1.75)	230.07 s (1.58)
Strategy 3	**66.82 s** (1.00)	76.25 s (1.00)	91.40 s (1.00)	145.82 s (1.00)

Table 4.4: Parallel execution time comparison between different initial guess strategies for the 2D nonlinear diffusion problem under a weak scalability approach with 9×10^4 degrees of freedom per subdomain and an overlap of $8h$. Values in parentheses represent the time ratio relative to the third strategy initial guess.

As a result, an additional computational cost is incurred due to the initialization of nonlinear local solvers with inappropriate values (such as zeros). This leads to additional nonlinear local iterations compared to RASPEN. To address this issue, a simple and inexpensive adjustment

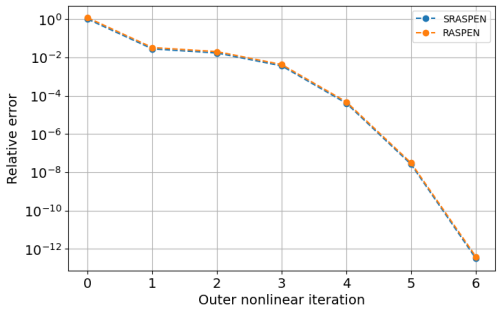


Fig. 4.6: Outer nonlinear convergence history of RASPEN and SRASPEN in the Forchheimer case with 20 subdomains, an overlap of $8h$, and a subdomain size of 200 degrees of freedom.

	RASPEN	**SRASPEN**
N = 16	57.38 s	56.40 s
N = 32	65.08 s	61.80 s
N = 64	108.10 s	87.49 s
N = 128	362.22 s	204.04 s

Table 4.5: Execution time comparison between SRASPEN and RASPEN in the 2D nonlinear diffusion problem under a weak scalability approach with 9.10^4 degrees of freedom per subdomain (tiles of 30×3000), an overlap of $8h$, and a unidirectional domain decomposition.

was proposed in this work to recover the iterates of RASPEN at the interior nodes within the SRASPEN process. This adjustment accelerates the subdomain nonlinear solves, resulting in a new variation that combines the high-accuracy iterates of RASPEN with the substructuring improvements of SRASPEN. Several numerical experiments showed the gains of the proposed variant on classical problems from the literature. The latter were carried out on up to 128 subdomains. At this scale, the use of a second-level preconditioner is often recommended to ensure the scalability of the method [12]. This issue will be addressed in a future work.

Acknoledgements. The authors would like to acknowledge the financial support provided by the Convention Industrielle de Formation par la Recherche (CIFRE) program (n°2022/1695), managed by the Association Nationale de Recherche et Technologie (ANRT), in collaboration with EDF.

References.

[1] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. *PETSc Users Manual*. Tech. rep. ANL-95/11 - Revision 3.6. Argonne National Laboratory, 2015.

[2] P. R. Brune, M. G. Knepley, B. F. Smith, and X. Tu. "Composing Scalable Nonlinear Algebraic Solvers". In: *SIAM Review* 57.4 (2015), pp. 535–565. DOI: 10.1137/130936725.

[3] X.-c. Cai and D. Keyes. "Nonlinearly Preconditioned Inexact Newton Algorithms". In: *SIAM Journal on Scientific Computing* 24 (Aug. 2001). DOI: 10.1137/S106482750037620X.

[4] F. Chaouqui, M. J. Gander, P. M. Kumbhar, and T. Vanzan. *Linear and nonlinear substructured Restricted Additive Schwarz iterations and preconditioning*. 2021.

[5] F. Chaouqui, M. J. Gander, P. M. Kumbhar, and T. Vanzan. "On the nonlinear Dirichlet-Neumann method and preconditioner for Newton's method". In: *ArXiv* abs/2103.12203 (2021).

[6] Z. Chen, G. Huan, and Y. Ma. *Computational Methods for Multiphase Flows in Porous Media*. SIAM, 2006.

[7] L. Dalcin, P. Kler, R. Paz, and A. Cosimo. "Parallel Distributed Computing using Python". In: *Advances in Water Resources* 34.9 (2011), pp. 1124–1139. DOI: 10.1016/j.advwatres.2011.04.013.

[8] R. S. Dembo, S. C. Eisenstat, and T. Steihaug. "Inexact Newton methods". In: *SIAM Journal on Numerical Analysis* 19 (1982), pp. 400–408.

[9] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[10] C. R. Dohrmann. "A Preconditioner for Substructuring Based on Constrained Energy Minimization". In: *SIAM Journal on Scientific Computing* 25.1 (2003), pp. 246–258.

[11] V. Dolean, M. J. Gander, W. Kheriji, F. Kwok, and R. Masson. "Nonlinear Preconditioning: How to Use a Nonlinear Schwarz Method to Precondition Newton's Method". In: *SIAM Journal on Scientific Computing* 38.6 (2016), A3357–A3380.

[12] V. Dolean, P. Jolivet, and F. Nataf. *An Introduction to Domain Decomposition Methods*. Society for Industrial and Applied Mathematics, 2015.

[13] M. Dryja and W. Hackbusch. "On the nonlinear domain decomposition method". In: *BIT Numerical Mathematics* 37 (1997), pp. 296–311.

[14] S. C. Eisenstat and H. F. Walker. "Globally convergent inexact Newton methods". In: *SIAM Journal on Optimization* 4 (1994), pp. 393–422.

[15] C. Farhat and F.-X. Roux. "A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm". In: *International Journal for Numerical Methods in Engineering* 32.6 (1991), pp. 1205–1227.

[16] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. "FETI-DP: a dual–primal unified FETI method—part I: A faster alternative to the two-level FETI method". In: *International Journal for Numerical Methods in Engineering* 50.7 (2001), pp. 1523–1544.

[17] P. Forchheimer. "Wasserbewegung durch Boden". In: *Zeitschrift des Vereines Deutscher Ingenieure* 45 (1901), pp. 1782–1788.

[18] S. Gong and X. Cai. "A Nonlinear Elimination Preconditioned Newton Method with Applications in Arterial Wall Simulation". In: *International Conference on Domain Decomposition Methods*. Springer, 2017, pp. 353–361.

[19] D. A. Ham, P. H. J. Kelly, L. Mitchell, C. J. Cotter, R. C. Kirby, K. Sagiyama, N. Bouziani, S. Vorderwuelbecke, T. J. Gregory, J. Betteridge, D. R. Shapero, R. W. Nixon-Hill, C. J. Ward, P. E. Farrell, P. D. Brubeck, I. Marsden, T. H. Gibson, M. Homolya, T. Sun, A. T. T. McRae, F. Luporini, A. Gregory, M. Lange, S. W. Funke, F. Rathgeber, G.-T. Bercea, and G. R. Markall. *Firedrake User Manual*. First edition. Imperial College London et al. May 2023. DOI: 10.25561/104839.

[20] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Philadelphia: SIAM, 1995.

[21] A. Klawonn, M. Lanser, and O. Rheinbach. "Nonlinear FETI-DP and BDDC Methods". In: *SIAM Journal on Scientific Computing* 36.2 (2014), A737–A765.

[22] A. Klawonn, M. Lanser, O. Rheinbach, and M. Uran. "Nonlinear FETI-DP and BDDC Methods: A Unified Framework and Parallel Results". In: *SIAM Journal on Scientific Computing* 39.6 (2017), pp. C417–C451.

[23] P. J. Lanzkron, D. J. Rose, and J. T. Wilkes. "An Analysis of Approximate Nonlinear Elimination". In: *SIAM Journal on Scientific Computing* 17.2 (1996), pp. 538–559.

[24] L. Liu, F.-N. Hwang, L. Luo, X.-C. Cai, and D. E. Keyes. "A Nonlinear Elimination Preconditioned Inexact Newton Algorithm". In: *SIAM Journal on Scientific Computing* 44.3 (2022), A1579–A1605.

[25] L. Luo, W.-S. Shiu, R. Chen, and X.-C. Cai. "A nonlinear elimination preconditioned inexact Newton method for blood flow problems in human artery with stenosis". In: *Journal of Computational Physics* 399 (2019), p. 108926. ISSN: 0021-9991.

[26] J. Mandel. "Balancing domain decomposition". In: *Communications in Numerical Methods in Engineering* 9.3 (1993), pp. 233–241.

[27] T. Mathew. *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations*. Vol. 61. Lecture Notes in Computational Science and Engineering. Springer, 2008.

[28] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Society for Industrial and Applied Mathematics, 2000.

[29] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations*. Numerical mathematics and scientific computation. Clarendon Press, 1999. ISBN: 9780198501787.

[30] Y. Saad and M. H. Schultz. "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems". In: *SIAM Journal on Scientific and Statistical Computing* 7.3 (July 1986), pp. 856–869. ISSN: 0196-5204.

[31] A. Toselli and O. Widlund. *Domain Decomposition Methods – Algorithms and Theory*. Vol. 34. Jan. 2005. ISBN: 978-3-540-20696-5.

[32] J. C. Ward. "Turbulent Flow in Porous Media". In: *Journal of the Hydraulics Division, ASCE* 90 (1964), pp. 1–12.

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399