

NONLINEAR FETI-DP AND BDDC METHODS: A UNIFIED FRAMEWORK AND PARALLEL RESULTS*

AXEL KLAWONN[†], MARTIN LANSER[†], OLIVER RHEINBACH[‡], AND MATTHIAS URAN[†]

Abstract. Parallel Newton–Krylov FETI-DP (Finite Element Tearing and Interconnecting—Dual-Primal) domain decomposition methods are fast and robust solvers, e.g., for nonlinear implicit problems in structural mechanics. In these methods, the nonlinear problem is first linearized and then decomposed into loosely coupled (linear) problems, which can be solved in parallel. By changing the order of the operations, new parallel communication can be constructed, where the loosely coupled local problems are nonlinear. We discuss different nonlinear FETI-DP methods which are equivalent when applied to linear problems but which show a different performance for nonlinear problems. Moreover, a new unified framework is introduced which casts all nonlinear FETI-DP domain decomposition approaches discussed in the literature into a single algorithm. Furthermore, the equivalence of nonlinear FETI-DP methods to specific nonlinearly right-preconditioned Newton–Krylov methods is shown. For the methods using nested Newton iterations, a strategy is presented to stop the inner Newton iteration early, resulting in an approximate local nonlinear elimination. Additionally, the nonlinear BDDC (Balancing Domain Decomposition by Constraint) method is presented as a right-preconditioned Newton approach. Finally, for the first time, parallel weak scaling results for four different nonlinear FETI-DP approaches are compared to standard Newton–Krylov FETI-DP in two and three dimensions, using both exact as well as highly scalable inexact linear FETI-DP preconditioners and up to 131 072 message passing interface (MPI) ranks on the JUQUEEN supercomputer at Forschungszentrum Jülich. For a model problem with nonlocal nonlinearities, nonlinear FETI-DP methods are shown to be up to five times faster than the standard Newton–Krylov FETI-DP approach.

Key words. nonlinear FETI-DP, nonlinear domain decomposition, nonlinear elimination, Newton’s method, nonlinear problems, parallel computing

AMS subject classifications. 68W10, 68U20, 65N55, 65F08, 65Y05

DOI. 10.1137/16M1102495

1. Introduction. Nonlinear domain decomposition methods (DDMs) are solution methods for nonlinear problems where the decomposition into a number of (loosely coupled) nonlinear problems on subdomains is performed before linearization. This is opposed to traditional methods, where the problem is first linearized and then decomposed into problems on subdomains. Reversing the order of linearization and decomposition localizes the computational work and reduces the need for synchro-

*Submitted to the journal’s Software and High-Performance Computing section November 8, 2016; accepted for publication (in revised form) August 28, 2017; published electronically November 16, 2017. Preliminary ideas on developing a nonlinear partial elimination framework for FETI-DP together with first sequential MATLAB experiments in two dimensions for Nonlinear FETI-DP-3, Nonlinear FETI-DP-4, and NL-ane- k , $k = 2, 3, 4$, have been already presented in the proceedings papers [35, 34]. The present paper significantly extends those preliminary ideas to a unified framework and also contains other new algorithmic and numerical results.

<http://www.siam.org/journals/sisc/39-6/M110249.html>

Funding: This work was supported in part by Deutsche Forschungsgemeinschaft (DFG) through the Priority Programme 1648 “Software for Exascale Computing” (SPPEXA) under grants KL 2094/4-1, KL 2094/4-2, RH 122/2-1, and RH 122/3-2.

[†]Mathematisches Institut, Universität zu Köln, Weyertal 86-90, 50931 Köln, Germany (axel.klawonn@uni-koeln.de, martin.lanser@uni-koeln.de, m.uran@uni-koeln.de, <http://www.numerik.uni-koeln.de>).

[‡]Institut für Numerische Mathematik und Optimierung, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, Akademiestr. 6, 09596 Freiberg, Germany (oliver.rheinbach@math.tu-freiberg.de, <http://www.mathe.tu-freiberg.de/nmo/mitarbeiter/oliver-rheinbach>).

nization on modern parallel computers. As in linear DDMs, the nonlinear methods typically need to incorporate a coarse problem to obtain scalability with respect to the number of subdomains. Among such methods are the ASPIN (Additive Schwarz Preconditioned Inexact Newton) method [9, 41, 25, 10, 24, 22, 21, 20], its multiplicative version MSPIN [45], RASPEN (Restricted Additive Schwarz Preconditioned Exact Newton) [15], nonlinear FETI-1 methods [50], and nonlinear Neumann–Neumann methods [5].

We are concerned with nonlinear versions of the successful family of (linear) FETI-DP (Finite Element Tearing and Interconnecting—Dual-Primal) [18, 17, 38, 39, 37, 40] and (linear) BDDC (Balancing Domain Decomposition by Constraint) [14, 12, 46, 44, 47] methods. Nonlinear BDDC and FETI-DP methods were introduced in [28, 27]. BDDC and FETI-DP methods are known for their robustness and parallel scalability: in 2009, weak parallel scalability was achieved for more than 65 000 cores in [38] for (linear) FETI-DP. Very recently, in [1], weak parallel scalability was achieved for almost half a million cores for a (linear) multilevel BDDC method. For nonlinear FETI-DP methods, weak parallel scalability was already obtained (for nonlinear elasticity problems) for more than half a million processor cores in [31], and, subsequently, for almost 800 000 processor cores [32]. This is currently the largest range of scalability reported for any linear or nonlinear DDM.

Our approach to nonlinear preconditioning separates the two goals of preconditioning for nonlinear problems. First, to improve the convergence of Newton’s method, the preconditioned nonlinear operator should be close to a function for which Newton’s method converges quickly and with a large convergence radius. Here, nonlinear preconditioning can thus also play a role similar to that of a globalization strategy. But even if the nonlinear preconditioner is chosen such that the preconditioned nonlinear operator is close to a linear operator (resulting in Newton convergence in a single iteration), this operator (and thus the tangent in Newton’s method) may still be ill-conditioned. Hence, second, an additional linear preconditioner can be chosen to obtain good convergence of the Krylov subspace iterative method used to compute the Newton correction. If an additional standard globalization strategy is used (e.g., a Trust-Region-Newton method or Newton’s method combined with gradient descent globalization), then the nonlinear preconditioner should enlarge the region where the Newton step is accepted. Let us note that, for such methods, the conditioning does influence the convergence speed of the gradient descent steps, and the smallest eigenvalue is related to the diameter of the region where the transition to Newton’s method occurs [53, proofs of Theorems 10.14 and 14.14].

Therefore, here, the convergence of the globalized Newton method and of the Krylov method are not completely unrelated.

In this paper, we review different versions of nonlinear FETI-DP methods. We present them in a common framework, which can help us better understand their respective performance for different nonlinear problems, and, later, test them making use of a common software framework built on PETSc [3, 4].

2. A unified framework for nonlinear FETI-DP. In this section, we introduce a new unified framework for the family of nonlinear FETI-DP methods, introduced in [28] and extended in [31, 35], and thus a common notation for all variants of nonlinear FETI-DP algorithms. In recent years, we have investigated variants with different nonlinear elimination strategies as well as different strategies to solve the linearized systems with exact and inexact linear FETI-DP methods; see [27, 30, 31, 28, 33, 35]. All these methods can now be described as a single non-

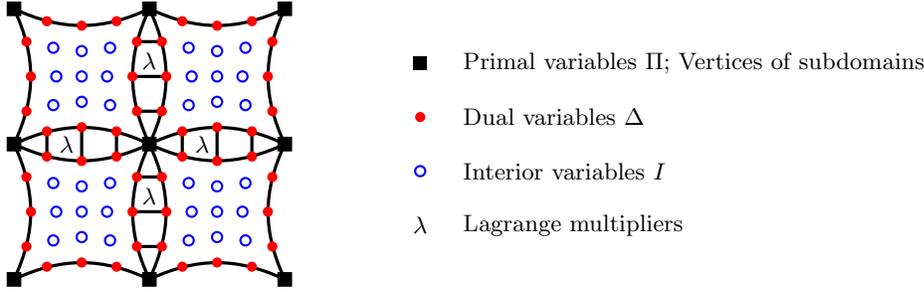


FIG. 1. Coupling of nonlinear local problems in the primal variables Π (black squares). Continuity in the dual interface variables Δ (red dots) is enforced by Lagrange multipliers λ . The remaining interior variables I are represented by blue circles. The nonlinear preconditioning in NL-4 performs exclusively on the interior variables (blue circles). In NL-3, the nonlinear preconditioning additionally executes on the dual interface variables (red dots). For NL-2 the nonlinear preconditioning performs on all variables, which includes the primal variables (black squares).

linear FETI-DP algorithm using different nonlinear right-preconditioners M , which describe nonlinear elimination processes. We hope that this can help to better understand the performance behavior of the different methods for various problem settings; see also section 5.

2.1. Spaces. Before we start with an abstract formulation of our framework let us briefly recall the basic aspects of FETI-DP. The computational domain Ω is geometrically subdivided into N nonoverlapping subdomains Ω_i , $i = 1, \dots, N$, and every subdomain is discretized by finite elements. The corresponding local finite element spaces are denoted $W^{(i)}$, and the product space $W := W^{(1)} \times \dots \times W^{(N)}$. The variables of the local solution vector $u^{(i)}$ can be partitioned into interior variables $u_I^{(i)}$, dual interface variables $u_\Delta^{(i)}$, and primal interface variables $u_\Pi^{(i)}$. The union of interior and dual variables is usually referred to as $u_B^{(i)} = (u_I^{(i)}, u_\Delta^{(i)})$. The local right-hand sides $f^{(i)}$ can be partitioned accordingly.

We also introduce the finite element space $\widetilde{W} \subset W$ of finite element functions, which are continuous in the primal variables (subdomain vertices or edge-constraints). The Schur complement in the primal variables represents the coarse operator of FETI-DP methods (see (19) and [52]) responsible for the global transport of information. We have $\tilde{u} = (u_B, \tilde{u}_\Pi) = (u_B^{(1)}, \dots, u_B^{(N)}, \tilde{u}_\Pi) \in \widetilde{W}$ and $\tilde{K}(\tilde{u}) \in \widetilde{W}$, where \tilde{K} is obtained from local subdomain operators by introduction of coupling (“glueing”) in the primal variables \tilde{u}_Π on the interface (see [28]) as in linear FETI-DP methods. Here, the primal interface variables \tilde{u}_Π are obtained from the local variables $u_\Pi^{(i)}$ by assembly. The remaining dual interface variables are denoted $u_\Delta = (u_\Delta^{(1)}, \dots, u_\Delta^{(N)})$, and the variables in the interior of subdomains are denoted $u_I = (u_I^{(1)}, \dots, u_I^{(N)})$; see also Figure 1. Analogously, we can assemble the local right-hand sides $f^{(i)} \in W^{(i)}$ to $\tilde{f} \in \widetilde{W}$. Furthermore, the space of Lagrange multipliers is defined as $V := \text{range}(B)$, where B is the standard linear FETI-DP jump operator and $\lambda \in V$.

2.2. Abstract formulation. Nonlinear FETI-DP methods are defined as iterative methods to solve the nonlinear system

$$(1) \quad A(\tilde{u}, \lambda) := \begin{bmatrix} \tilde{K}(\tilde{u}) + B^T \lambda - \tilde{f} \\ B\tilde{u} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

1. Mapping: $M : \widetilde{W} \times V \rightarrow \widetilde{W} \times V$.
2. M puts the current iterate into the neighborhood of the solution; see also [7].
3. $M(\tilde{u}, \lambda)$ is easily computable compared to the inverse action of $A(\tilde{u}, \lambda)$.

FIG. 2. *Properties of the nonlinear preconditioner M for nonlinear FETI-DP methods.*

see [28]. The (sparse) linear constraint $B\tilde{u} = 0$ enforces the continuity of the solution across the subdomain interface for the nonprimal variables u_B ; see [28] for the notation and the details. Lagrange multipliers λ are used to decompose the nonlinear problem into loosely coupled local nonlinear problems on subdomains. We always assume that the coupling of the nonlinear subdomain problems introduced by the primal variables \tilde{u}_Π is sufficient to make the local problems (and the Jacobian $D\tilde{K}$) invertible. Such a set of primal variables can always be found if the original, undecomposed problem is invertible, and, importantly, the set is typically small. We denote the exact solution of (1) by (\tilde{u}^*, λ^*) . Usually, we obtain the system (1) from the minimization of a nonlinear (e.g., hyperelastic) energy on the subdomains under a continuity constraint on the subdomain interfaces; see [28, equation (2.4)].

Next, instead of simply linearizing (1), we first apply a nonlinear right-preconditioner (i.e., inner preconditioner) $M(\tilde{u}, \lambda)$ to the system (1). Subsequently, $A(M(\tilde{u}, \lambda))$ is linearized using Newton's method. Our choices of $M(\tilde{u}, \lambda)$ can be characterized as different nonlinear elimination processes [43] and can be viewed in the context of nonlinear right-preconditioning [11, 8] in Newton's method. In fact, directly linearizing (1), i.e., choosing $M(\tilde{u}, \lambda)$ as the identity, leads to the method Nonlinear-FETI-DP-1; see section 2.5.5. Finally, in each Newton step, the linearized system is solved iteratively by a Krylov subspace method such as CG or GMRES following the classical (linear) FETI-DP approach [52], and applying a (linear) Dirichlet-type preconditioner [52] during the Krylov iteration.

We now describe different nonlinear FETI-DP methods [28, 27, 35] as solving the nonlinear equation

$$(2) \quad A(M(\tilde{u}, \lambda)) = 0$$

by an iterative scheme, e.g., typically by a Newton–Krylov method.

Linearizing in variables (\tilde{u}, λ) when applying Newton's method to (2), we obtain the iteration

$$(3) \quad \begin{bmatrix} \tilde{u}^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} := \begin{bmatrix} \tilde{u}^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \alpha^{(k)} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix}$$

with a suitable step length $\alpha^{(k)}$ and, from the chain rule, the Newton update

$$(4) \quad \left(DA(M(\tilde{u}^{(k)}, \lambda^{(k)})) \cdot DM(\tilde{u}^{(k)}, \lambda^{(k)}) \right) \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = A(M(\tilde{u}^{(k)}, \lambda^{(k)})).$$

In each Newton step, the nonlinear preconditioner is evaluated by computing

$$g^{(k)} := M(\tilde{u}^{(k)}, \lambda^{(k)}).$$

Note that the nonlinear preconditioner M should satisfy some properties or assumptions in order to accelerate the computations; see Figure 2.

Moreover, when solving (2), we are in fact not interested in obtaining \tilde{u}^{**} and λ^{**} satisfying $A(M(\tilde{u}^{**}, \lambda^{**})) = 0$, but, as in linear right-preconditioning, we are rather interested in $(\tilde{u}^*, \lambda^*) = M(\tilde{u}^{**}, \lambda^{**})$.

2.3. Local convergence analysis. The equivalence of the nonlinear FETI-DP saddle point system (1) and the fully assembled original finite element problem (e.g., (27)) has already been discussed in [28]. In this section, we provide a brief discussion of the convergence behavior of nonlinear FETI-DP methods.

ASSUMPTION 2.1. *Let U be an open neighborhood of our solution (\tilde{u}^*, λ^*) . We assume that $A(\tilde{u}, \lambda)$ is continuously differentiable in U and that $DA(\tilde{u}^*, \lambda^*)$ is a nonsingular matrix.*

If Assumption 2.1 is satisfied, Newton’s method solving $A(\tilde{u}, \lambda) = 0$ will converge for all initial values $(\tilde{u}^{(0)}, \lambda^{(0)}) \in U^* \subset U$ to the solution (\tilde{u}^*, λ^*) ; see, e.g., Ortega and Rheinboldt [49, section 10.2.2]. If $DA(\tilde{u}^*, \lambda^*)$ is nonsingular, then, by introducing a sufficient number of primal constraints, $D\tilde{K}$ is also nonsingular in u^* and a small neighborhood of u^* .

In the case of redundant Lagrange multipliers $DA(\tilde{u}^*, \lambda^*)$ will be singular since B will not have full rank. However, this will lead to the same Newton iterates $\tilde{u}^{(k)}$ and thus the same convergence behavior as when using nonredundant multipliers. More precisely, in remarks 1 and 2 in [42, section 2.5.2] it is proved by induction that the Newton iterates $(\tilde{u}_1^{(k)}, \lambda_1^{(k)})$, which nonlinear FETI-DP delivers for a system with nonredundant multipliers λ_1 and the corresponding jump matrix B_1 , have the relationship $\tilde{u}_1^{(k)} = \tilde{u}^{(k)}$ and $\lambda_1^{(k)} = R_{n_1} L^T \lambda^{(k)}$ to the iterates in the redundant case. Here, R_{n_1} is a simple restriction matrix, and L is a nonsingular lower triangular matrix fulfilling the property $B = L [B_1^T \ 0]^T$. Let us remark that of course $\lambda_1^{(0)} = R_{n_1} L^T \lambda^{(0)}$ has to be fulfilled for the initial value, which is the case for our choice $\lambda^{(0)} = 0$ and $\lambda_1^{(0)} = 0$. Note that stopping criteria involving the updates $\delta\lambda^{(k)}$ have to be avoided if redundant Lagrange multipliers are used, since in that case convergence can only be guaranteed in \tilde{W} . As mentioned above, we can guarantee $\tilde{u}_1^{(k)} = \tilde{u}^{(k)}$ and thus also $\tilde{u}_1^* = \tilde{u}^*$ for the solution in \tilde{W} , but since $R_{n_1} L^T$ has a nontrivial kernel we might obtain Newton updates $\delta\lambda^{(k)}$ with $\|\delta\lambda^{(k)}\| > 0$ in the case of redundant multipliers, even if convergence is already reached; see also [42, end of section 2.5.2].

ASSUMPTION 2.2. *Let V^* be an open neighborhood of (\tilde{u}^*, λ^*) . The function evaluation $M(\tilde{u}, \lambda)$ is well defined and computable in V^* and $M(V^*) \subset U^*$.*

Under Assumption 2.2 we have for all iterates $(\tilde{u}^{(k)}, \lambda^{(k)}) \in V^*$ that $M(\tilde{u}^{(k)}, \lambda^{(k)}) \in U^*$, and thus, with Assumption 2.1 and the discussion above, the nonlinear FETI-DP method converges for all initial values $(\tilde{u}^{(0)}, \lambda^{(0)}) \in V^*$ to the solution (\tilde{u}^*, λ^*) . If V^* is larger than U^* , the convergence radius is increased.

The computational cost for the nonlinear preconditioners has to be assessed for each M , separately, again relying on assumptions on local differentiability and invertibility. For NL-2 and thus for M defined as in (21), we already provided a discussion in [28].

2.4. Construction of four different variants. The choice of the nonlinear preconditioner $M(\tilde{u}, \lambda)$ is essential for the properties of the nonlinear FETI-DP method. We will consider four different choices for the nonlinear preconditioner M , i.e.,

- linear preconditioner M (Nonlinear-FETI-DP-1 (NL-1); see also [27, 30, 31, 28, 33]),
- nonlinear preconditioning of \tilde{u} (Nonlinear-FETI-DP-2 (NL-2); see also [28]),
- nonlinear preconditioning of u_B (Nonlinear-FETI-DP-3 (NL-3); see also [35]),
- nonlinear preconditioning of u_I (Nonlinear-FETI-DP-4 (NL-4); see also [35]).

For linear problems, all four methods are equivalent and basically reduce to the standard linear FETI-DP method.

We will now present all four methods in a common framework and later, in section 4, will consider an important modification which, in our experience, can improve the performance significantly.

2.5. Abstract formulation using partial nonlinear elimination.

2.5.1. Derivation of the method. We will now show how nonlinear FETI-DP methods can be defined using partial nonlinear elimination.

Note that we restrict ourselves to nonlinear preconditioners M which are nonlinear in \tilde{u} and linear in λ , i.e., we do not eliminate any Lagrange multipliers. Here, in fact, the preconditioner is only the identity in λ . Therefore, our M always has the form

$$(5) \quad M(\tilde{u}, \lambda) := (M_{\tilde{u}}(\tilde{u}, \lambda), \lambda).$$

Classical linear FETI-DP methods are based on a decomposition of the variable \tilde{u} into several subsets, such as, e.g., the decomposition of the interface variables u_Γ into dual variables u_Δ and primal variables \tilde{u}_Π .

The index sets can also be used for the construction of a nonlinear preconditioner M . We will first discuss the general approach of a variable splitting in the nonlinear context and then derive four different nonlinear FETI-DP variants, of which two turn out to be identical to methods already introduced in [35]. Since the effect of these nonlinear FETI-DP preconditioners can be interpreted as a partial nonlinear elimination process, corresponding sets of variables \tilde{u}_E and \tilde{u}_L are introduced. Here, the index E marks the set of variables which will be eliminated nonlinearly and L the set of variables which will be linearized. Thus, we define the splitting

$$\tilde{u} = (\tilde{u}_E, \tilde{u}_L).$$

Analogously, we partition the jump operator B (see (1))

$$B = [B_E \ B_L],$$

and our system (1) then reads

$$(6) \quad A(\tilde{u}_E, \tilde{u}_L, \lambda) = \begin{bmatrix} \tilde{K}_E(\tilde{u}_E, \tilde{u}_L) + B_E^T \lambda - \tilde{f}_E \\ \tilde{K}_L(\tilde{u}_E, \tilde{u}_L) + B_L^T \lambda - \tilde{f}_L \\ B_E \tilde{u}_E + B_L \tilde{u}_L \end{bmatrix} = 0.$$

We define a nonlinear preconditioner of the form

$$(7) \quad M_{\tilde{u}}(\tilde{u}, \lambda) := (M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda), \tilde{u}_L),$$

where $M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda)$ is defined implicitly by the first line of (6); cf. [35, eq. (5)]. The variable \tilde{u}_E is thus eliminated from (6) by solving the nonlinear equation

$$(8) \quad \tilde{K}_E(M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda), \tilde{u}_L) + B_E^T \lambda - \tilde{f}_E = 0$$

using Newton’s method; see subsection 2.5.2. Note that in fact $M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda) = M_{\tilde{u}_E}(\tilde{u}_L, \lambda)$; i.e., $M_{\tilde{u}_E}$ is independent of its first argument \tilde{u}_E , which we only introduce for convenience, such that $DM_{\tilde{u}_E}$ is a square matrix; see subsection 2.5.2. Replacing

\tilde{u}_E in the second and third lines of (6) by $M_{\tilde{u}_E}(\tilde{u}_L, \lambda)$, we obtain the nonlinear Schur complement

$$(9) \quad S_L(\tilde{u}_L, \lambda) := \begin{bmatrix} \tilde{K}_L(M_{\tilde{u}_E}(\tilde{u}_L, \lambda), \tilde{u}_L) + B_L^T \lambda - \tilde{f}_L \\ B_E M_{\tilde{u}_E}(\tilde{u}_L, \lambda) + B_L \tilde{u}_L \end{bmatrix}.$$

The Schur complement system

$$(10) \quad S_L(\tilde{u}_L, \lambda) = 0$$

can now be linearized, i.e., solved by Newton’s method. The tangent DS_L of S_L can be obtained from the chain rule and the implicit function theorem; see subsection 2.5.2. Note that the nonlinearly preconditioned system (2) takes the form

$$A(M(\tilde{u}, \lambda)) = A(M_{\tilde{u}}(\tilde{u}, \lambda), \lambda) = A(M_{\tilde{u}_E}(\tilde{u}_L, \lambda), \tilde{u}_L, \lambda) = \begin{pmatrix} 0 \\ S_L(\tilde{u}_L, \lambda) \end{pmatrix}.$$

Let us now gather the building blocks necessary to implement Newton’s method applied to (10).

2.5.2. Computing the tangent. For each application of the preconditioner M , a nonlinear system

$$\tilde{K}_E(g_E, \tilde{u}_L) + B_E^T \lambda - \tilde{f}_E = 0$$

has to be solved for g_E ; cf. (5), (7), and (8). For the computation of

$$(11) \quad g_E^{(k)} := M_{\tilde{u}_E}(\tilde{u}_E^{(k)}, \tilde{u}_L^{(k)}, \lambda^{(k)}),$$

Newton’s method can be applied and yields the iteration

$$(12) \quad g_{E,l+1}^{(k)} := g_{E,l}^{(k)} - \left(D\tilde{K}(g_{E,l}^{(k)}, \tilde{u}_L^{(k)})_{EE} \right)^{-1} \left(\tilde{K}_E(g_{E,l}^{(k)}, \tilde{u}_L^{(k)}) + B_E^T \lambda^{(k)} - \tilde{f}_E \right),$$

which converges to $g_E^{(k)}$ under sufficient assumptions always made throughout this paper. Here, k is the index of the outer Newton iteration, and l is the index of the inner Newton iteration.

Next, we assume the following partitioning of the tangent of \tilde{K} :

$$(13) \quad D\tilde{K}(\cdot) = \begin{bmatrix} D\tilde{K}(\cdot)_{EE} & D\tilde{K}(\cdot)_{EL} \\ D\tilde{K}(\cdot)_{LE} & D\tilde{K}(\cdot)_{LL} \end{bmatrix}.$$

Obviously, we obtain

$$(14) \quad g^{(k)} = M_{\tilde{u}}(\tilde{u}^{(k)}, \lambda^{(k)}) = \left(g_E^{(k)}, \tilde{u}_L^{(k)} \right).$$

We compute the derivative of $M_{\tilde{u}_E}$ from (8) with respect to the first variable \tilde{u}_E and obtain

$$D_{\tilde{u}_E} \tilde{K}_E(M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda), \tilde{u}_L) \cdot D_{\tilde{u}_E} M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda) = 0 \Leftrightarrow D_{\tilde{u}_E} M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda) = 0,$$

assuming invertibility of $D_{\tilde{u}_E} \tilde{K}_E$ (see subsection 2.3).

Computing in (8) the derivative with respect to \tilde{u}_L yields

$$D_{\tilde{u}_E} \tilde{K}_E(M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda), \tilde{u}_L) D_{\tilde{u}_L} M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda) + D_{\tilde{u}_L} \tilde{K}_E(M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda), \tilde{u}_L) = 0,$$

which is equivalent to

$$\begin{aligned} & D_{\tilde{u}_L} M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda) \\ &= - \left(D_{\tilde{u}_E} \tilde{K}_E(M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda), \tilde{u}_L) \right)^{-1} D_{\tilde{u}_L} \tilde{K}_E(M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda), \tilde{u}_L). \end{aligned}$$

Computing in (8) the derivative with respect to λ yields

$$\begin{aligned} D_{\tilde{u}_E} \tilde{K}_E(M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda), \tilde{u}_L) D_{\lambda} M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda) + B_E^T &= 0 \quad \text{or equivalently} \\ D_{\lambda} M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda) &= - \left(D_{\tilde{u}_E} \tilde{K}_E(M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda), \tilde{u}_L) \right)^{-1} B_E^T. \end{aligned}$$

Thus, the derivative of $M(\tilde{u}_E^{(k)}, \tilde{u}_L^{(k)}, \lambda^{(k)})$ (see (5) and (7)) with respect to $(\tilde{u}_E, \tilde{u}_L, \lambda)$ and using the notation from (13) is

$$\begin{aligned} & DM(\tilde{u}_E^{(k)}, \tilde{u}_L^{(k)}, \lambda^{(k)}) \\ &= \begin{bmatrix} 0 & -D\tilde{K}_{EE}^{-1}(g^{(k)})D\tilde{K}_{EL}(g^{(k)}) & -D\tilde{K}_{EE}^{-1}(g^{(k)})B_E^T \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}. \end{aligned}$$

Thus, the left-hand side of the Newton system (4) writes as

$$\begin{aligned} & DA(g^{(k)}, \lambda^{(k)}) \cdot DM(\tilde{u}^{(k)}, \lambda) \\ &= \begin{bmatrix} D\tilde{K}_{EE} & D\tilde{K}_{EL} & B_E^T \\ D\tilde{K}_{LE} & D\tilde{K}_{LL} & B_L^T \\ B_E & B_L & 0 \end{bmatrix} \begin{bmatrix} 0 & -D\tilde{K}_{EE}^{-1}D\tilde{K}_{EL} & -D\tilde{K}_{EE}^{-1}B_E^T \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & -D\tilde{K}_{LE}D\tilde{K}_{EE}^{-1}D\tilde{K}_{EL} + D\tilde{K}_{LL} & -D\tilde{K}_{LE}^{-1}D\tilde{K}_{EE}^{-1}B_E^T + B_L^T \\ 0 & -B_E D\tilde{K}_{EE}^{-1}D\tilde{K}_{EL} + B_L & -B_E D\tilde{K}_{EE}^{-1}B_E^T \end{bmatrix} \\ (15) \quad &=: \begin{bmatrix} 0 & 0 & 0 \\ 0 & DS_{LL} & DS_{L\lambda} \\ 0 & DS_{\lambda L} & DS_{\lambda\lambda} \end{bmatrix}; \end{aligned}$$

cf. [35, eq. (11)]. For better readability, we write $D\tilde{K}$ instead of $D\tilde{K}(g^{(k)})$ and define the operator

$$DS_L(g^{(k)}) := \begin{bmatrix} DS_{LL}(g^{(k)}) & DS_{L\lambda}(g^{(k)}) \\ DS_{\lambda L}(g^{(k)}) & DS_{\lambda\lambda}(g^{(k)}) \end{bmatrix}.$$

As a result of the chain rule and the implicit function theorem, the operator DS_L is, under sufficient conditions, the tangent of the nonlinear Schur complement S_L defined in (9).

As a result of (8) and (15), we can write the Newton update (4) as

$$(16) \quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & DS_{LL}(g^{(k)}) & DS_{L\lambda}(g^{(k)}) \\ 0 & DS_{\lambda L}(g^{(k)}) & DS_{\lambda\lambda}(g^{(k)}) \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_E^{(k)} \\ \delta\tilde{u}_L^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{K}_L(g^{(k)}) + B_L^T \lambda^{(k)} - \tilde{f}_L \\ Bg^{(k)} \end{bmatrix}.$$

The iterations defined in (16) and in [35, eq. (10)] are equivalent as a consequence of the chain rule and the implicit function theorem. This will later be sufficient to show

Init: $\tilde{u}^{(0)} \in \widetilde{W}$, $\lambda^{(0)} \in V$
Iterate over k :
Compute:
 $(g^{(k)}, \lambda^{(k)}) := (M_{\tilde{u}}(\tilde{u}^{(k)}, \lambda^{(k)}), \lambda^{(k)}) = M(\tilde{u}^{(k)}, \lambda^{(k)})$
 /* Often requires solution of localized nonlinear problems */
If $\|A(g^{(k)}, \lambda^{(k)})\|$ sufficiently small
break; /* Convergence of nonlinear FETI-DP; small absolute residual; */
Solve the linearized system by a Krylov iteration as in a standard linear FETI-DP approach:
 $(DA(g^{(k)}, \lambda^{(k)}) \cdot DM(\tilde{u}^{(k)}, \lambda^{(k)})) \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = A(g^{(k)}, \lambda^{(k)})$
Update: $\tilde{u}^{(k+1)} := \tilde{u}^{(k)} - \alpha^{(k)}\delta\tilde{u}^{(k)}$
Update: $\lambda^{(k+1)} := \lambda^{(k)} - \alpha^{(k)}\delta\lambda^{(k)}$
End Iteration

FIG. 3. Nonlinear FETI-DP algorithm(s). We always use $\tilde{u}^{(k+1)}$ as initial value for the computation of $g^{(k+1)}$.

that the methods defined in subsection 2.5.6, subsection 2.5.7, and subsection 2.5.8 below as NL-2, NL-3, and NL-4 are identical to the methods defined in [28, 27] and [35].

Note that $M(\tilde{u}^{(k)}, \lambda^{(k)})$ is independent of $\tilde{u}_E^{(k)}$; however, we use $\tilde{u}_E^{(k+1)}$ as an initial value for the computation of $g^{(k+1)}$; see Figure 3. Note that in efficient implementations the Schur complement DS_L is never assembled and that eliminating the block $D\tilde{K}_{EE}(g^{(k)})$ from $DA(g^{(k)}, \lambda^{(k)})$ also leads to the Schur complement system (16). Therefore, we typically replace the left-hand side in (16) by $DA(g^{(k)}, \lambda^{(k)})$; i.e., we will solve

$$(17) \quad \begin{bmatrix} D\tilde{K}_{EE} & D\tilde{K}_{EL} & B_E^T \\ D\tilde{K}_{LE} & D\tilde{K}_{LL} & B_L^T \\ B_E & B_L & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_E^{(k)} \\ \delta\tilde{u}_L^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{K}_L(g^{(k)}) + B_L^T \lambda^{(k)} - \tilde{f}_L \\ Bg^{(k)} \end{bmatrix};$$

see subsection 2.5.6, subsection 2.5.7, and subsection 2.5.8 for details. This does not affect the updates $\delta\tilde{u}_L^{(k)}$ and $\delta\lambda^{(k)}$, but we obtain $\delta\tilde{u}_E^{(k)}$, which can be useful, and, moreover, inexact or inexact reduced FETI-DP methods can be constructed; see subsection 2.6.

Now, we can summarize the different nonlinear FETI-DP methods described in [27, 30, 31, 28, 33, 35] as a single algorithm; see Figure 3.

2.5.3. Some algorithmic details. Throughout this paper, we assume that the primal space \widetilde{W} can be chosen such that $D\tilde{K}(\tilde{u})$ is invertible; see above. If that is not the case for an iterate $\tilde{u}^{(k)}$, regularization may be necessary, or a gradient descent step may replace the Newton step, as in standard globalization approaches.

In all nonlinear FETI-DP methods presented here, we have to solve systems with a left-hand side of the type

$$\begin{bmatrix} D\tilde{K}(\tilde{u}) & B^T \\ B & 0 \end{bmatrix},$$

where $D\tilde{K}(\tilde{u})$ can be sorted as follows:

$$D\tilde{K}(\tilde{u}) = \begin{bmatrix} D\tilde{K}(\tilde{u})_{BB} & D\tilde{K}(\tilde{u})_{B\Pi} \\ D\tilde{K}(\tilde{u})_{\Pi B} & D\tilde{K}(\tilde{u})_{\Pi\Pi} \end{bmatrix}.$$

As before, the index set Π denotes the primal variables, and the index set B denotes the union of the variables in the remaining interface (dual variables) and on the interior of the subdomains. As in standard FETI-DP methods, the matrix $D\tilde{K}(\tilde{u})_{BB}$ has the block diagonal structure

$$(18) \quad D\tilde{K}(\tilde{u})_{BB} = \begin{bmatrix} D\tilde{K}^{(1)}(\tilde{u}^{(1)})_{BB} & & \\ & \ddots & \\ & & D\tilde{K}^{(N)}(\tilde{u}^{(N)})_{BB} \end{bmatrix},$$

suitable for parallelization, where each block $D\tilde{K}^{(i)}(\tilde{u}^{(i)})_{BB}$, $i = 1, \dots, N$, corresponds to a single subdomain.

In nonlinear FETI-DP methods with exact solvers (see [27, 28]), a sparse factorization of $D\tilde{K}(\tilde{u})$ is performed by first carrying out parallel, local, sparse factorizations of the diagonal blocks $D\tilde{K}^{(1)}(\tilde{u}^{(1)})_{BB}, \dots, D\tilde{K}^{(N)}(\tilde{u}^{(N)})_{BB}$ and, subsequently, by a factorization of the primal Schur complement

$$(19) \quad \tilde{S}_{\Pi\Pi} := D\tilde{K}(\tilde{u})_{\Pi\Pi} - D\tilde{K}(\tilde{u})_{\Pi B} D\tilde{K}(\tilde{u})_{BB}^{-1} D\tilde{K}(\tilde{u})_{B\Pi},$$

which constitutes the FETI-DP coarse operator. In inexact and inexact reduced nonlinear FETI-DP methods [31, 30, 33] an exact factorization of $\tilde{S}_{\Pi\Pi}$ can be avoided. Instead, an algebraic multigrid (AMG) preconditioner is set up for $\tilde{S}_{\Pi\Pi}$; see subsection 2.6.

2.5.4. Common approximation in right-preconditioned Newton. In right-preconditioned Newton–Krylov the computation of DM^{-1} is sometimes avoided by using a first order approximation of M (see [7]); i.e., instead of

$$\begin{aligned} & \begin{bmatrix} \tilde{u}^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} \\ = & \begin{bmatrix} \tilde{u}^{(k)} \\ \lambda^{(k)} \end{bmatrix} - \alpha^{(k)} \left(DM(\tilde{u}^{(k)}, \lambda^{(k)}) \right)^{-1} \left(DA(\tilde{u}^{(k+\frac{1}{2})}, \lambda^{(k+\frac{1}{2})}) \right)^{-1} A(\tilde{u}^{(k+\frac{1}{2})}, \lambda^{(k+\frac{1}{2})}) \end{aligned}$$

the iteration

$$\begin{aligned} \begin{bmatrix} \tilde{u}^{(k+\frac{3}{2})} \\ \lambda^{(k+\frac{3}{2})} \end{bmatrix} &= M(\tilde{u}^{(k+1)}, \lambda^{(k+1)}) \\ &\approx M(\tilde{u}^{(k)}, \lambda^{(k)}) - \alpha^{(k)} \left(DA(\tilde{u}^{(k+\frac{1}{2})}, \lambda^{(k+\frac{1}{2})}) \right)^{-1} A(\tilde{u}^{(k+\frac{1}{2})}, \lambda^{(k+\frac{1}{2})}) \\ \begin{bmatrix} \tilde{u}^{(k+1)} \\ \lambda^{(k+1)} \end{bmatrix} &= M^{-1}(\tilde{u}^{(k+\frac{3}{2})}, \lambda^{(k+\frac{3}{2})}) \end{aligned}$$

is used. This approach utilizes that the application (or approximation) of M^{-1} is usually cheap. We, however, usually do not avoid DM^{-1} : As a result of (2), our methods can be seen as right-preconditioned Newton methods [7], where the tangent is computed exactly and not approximately; see (4). However, in section 4, we will approximate the action of M^{-1} .

2.5.5. Nonlinear-FETI-DP-1. We will now describe the four specific variants of nonlinear FETI-DP. The NL-1 and NL-2 methods [28, 27] will be described first; they constitute the two extreme cases of partial nonlinear elimination, i.e., no elimination (NL-1) and full elimination (NL-2).

We obtain Nonlinear-FETI-DP-1 (NL-1), as introduced in [28, 27], by choosing the index sets $E = \emptyset$ and $L = [I \ \Delta \ \Pi]$. Then, M is the identity

$$M(\tilde{u}, \lambda) := (\tilde{u}, \lambda),$$

and $g^{(k)} := M_{\tilde{u}}(\tilde{u}^{(k)}, \lambda^{(k)})$ (see (11)) reduces to $g^{(k)} := \tilde{u}^{(k)}$.

We obtain $DM(\tilde{u}^{(k)}, \lambda^{(k)}) = I$, and thus the linearized system (4) writes as

$$(20) \quad \begin{bmatrix} D\tilde{K}(\tilde{u}^{(k)}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} \tilde{K}(\tilde{u}^{(k)}) + B^T\lambda^{(k)} - \tilde{f} \\ B\tilde{u}^{(k)} \end{bmatrix}.$$

Equation (20) can be solved as in any inexact or exact FETI-DP-type method using a standard (linear) preconditioner for the dual Schur complement. Thus, we can solve

$$B \left(D\tilde{K}(g^{(k)}) \right)^{-1} B^T \delta\lambda^{(k)} = -B\tilde{u}^{(k)} + B(D\tilde{K}(\tilde{u}^{(k)})^{-1})(\tilde{K}(\tilde{u}^{(k)}) + B^T\lambda^{(k)} - \tilde{f})$$

by a Krylov method using the standard FETI-DP Dirichlet preconditioner M_D [52].

As is standard, the operator $(D\tilde{K}(g^{(k)}))^{-1}$ is never formed explicitly, but its application to a vector is computed by using parallel, local sparse LU-factorizations and the solution of a small, globally coupled coarse Schur complement; cf. (18) and (19), and also [28]. For more details, see subsection 2.5.3.

This method is the Nonlinear-FETI-DP-1 method (see [28, eq. (3.5)] or [27, eq. (4)]). In practice, NL-1 often performs similarly to classical Newton–Krylov–FETI-DP. The method can be improved by using an initial value computed from solving the nonlinear problem $\tilde{K}(\tilde{u}) = \tilde{f} - B^T\lambda$; see [27, 30, 31, 28]. In our numerical results, we always include the computation of this initial value.

2.5.6. Nonlinear-FETI-DP-2. We obtain Nonlinear-FETI-DP-2 (NL-2), as introduced in [28, 27], by choosing the splitting $E = [I \ \Delta \ \Pi]$ and $L = \emptyset$. Thus, the preconditioner $M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda) = M_{\tilde{u}}(\tilde{u}, \lambda)$ is defined implicitly by

$$(21) \quad \tilde{K}(M_{\tilde{u}}(\tilde{u}, \lambda)) + B^T\lambda - \tilde{f} = 0;$$

see (8). The computation of $g^{(k)} := M_{\tilde{u}}(\tilde{u}^{(k)}, \lambda^{(k)})$ (see (11)) is performed by applying Newton's method, which yields the iteration

$$(22) \quad g_{l+1}^{(k)} := g_l^{(k)} - \left(D\tilde{K}(g_l^{(k)}) \right)^{-1} \left(\tilde{K}(g_l^{(k)}) + B^T\lambda^{(k)} - \tilde{f} \right),$$

assumed to converge to $g^{(k)}$; cf. (12). Here, k is the index of the outer Newton iteration (see (3), (4), and also Figure 3), and l is the index of the inner Newton iteration to compute $g^{(k)}$, which is needed to compute the right-hand side of (4).

For NL-2, the system (17) reads

$$(23) \quad \begin{bmatrix} D\tilde{K}(\tilde{u}^{(k)}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ B\tilde{u}^{(k)} \end{bmatrix}.$$

Again, we can now solve the Schur complement system

$$(24) \quad B \left(D\tilde{K}(g^{(k)}) \right)^{-1} B^T \delta\lambda^{(k)} = -Bg^{(k)}$$

using a Krylov method (and the standard FETI-DP Dirichlet preconditioner [52]) and setting $\delta\tilde{u}^{(k)} := 0$. This iteration, defined by (22) and (24), is equivalent to the Nonlinear-FETI-DP-2 algorithm; cf. [28, eq. (3.17), (3.18), and (3.19)].

As observed in subsection 2.5, in practice, we propose solving (23) instead of (24). Then, the update for the Lagrange multipliers $\delta\lambda^{(k)}$ is identical, but we additionally obtain $\delta\tilde{u}^{(k)}$, which can be used to update the initial value for the computation of $g^{(k+1)}$. We can also apply inexact or inexact reduced FETI-DP methods without changing the solution; see subsection 2.6.

The NL-2 approach corresponds to an exact nonlinear elimination of \tilde{u} , and we have an outer and an inner Newton iteration; i.e., whenever a residual is evaluated in the outer Newton iteration, a nonlinear system has to be solved by an inner Newton iteration.

The next two nonlinear FETI-DP methods, NL-3 and NL-4, also have an inner and an outer Newton iteration, but the inner Newton iteration is cheaper since, in both cases, it does not include the coarse problem and its all-to-all communication.

2.5.7. Nonlinear-FETI-DP-3. We obtain Nonlinear-FETI-DP-3 (NL-3), as introduced in [35], by choosing $E := B = [I \ \Delta]$ and $L := \Pi$. The preconditioner $M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda)$ is defined implicitly in (8). Thus, we immediately obtain $g^{(k)} = (g_B^{(k)}, \tilde{u}_\Pi^{(k)})$ and

$$g_{B,l+1}^{(k)} := g_{B,l}^{(k)} - \left(D\tilde{K}(g_B^{(l)}, \tilde{u}_\Pi^{(l)})_{BB} \right)^{-1} \left(\tilde{K}_B(g_{B,l}^{(k)}, \tilde{u}_\Pi^{(k)}) + B_B^T \lambda^{(k)} - \tilde{f}_B \right)$$

from (12) and (14), converging to $g_B^{(k)}$; cf. [35, eq. (13)].

Since we have continuity in all primal variables, we can assume that $B_L := B_\Pi = 0$.

Thus, the linearized FETI-DP system writes as

$$(25) \quad \begin{bmatrix} D\tilde{K}(g^{(k)})_{BB} & D\tilde{K}(g^{(k)})_{B\Pi} & B_B^T \\ D\tilde{K}(g^{(k)})_{\Pi B} & D\tilde{K}(g^{(k)})_{\Pi\Pi} & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_B^{(k)} \\ \delta\tilde{u}_\Pi^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{K}_\Pi(g^{(k)}) - \tilde{f}_\Pi \\ Bg^{(k)} \end{bmatrix}.$$

In NL-3, local nonlinear problems in the variable \tilde{u}_B have to be solved. The resulting computational work is completely local and also does not involve any operations on the FETI-DP coarse space. This property offers the potential to reduce the number of primal assembly processes and FETI-DP coarse solves, which can lead to improved scalability.

2.5.8. Nonlinear-FETI-DP-4. We obtain Nonlinear-FETI-DP-4 (NL-4), as introduced in [35], by choosing $E := I$ and $L := [\Delta \ \Pi]$. The preconditioner $M_{\tilde{u}_E}(\tilde{u}_E, \tilde{u}_L, \lambda)$ is defined implicitly by (8). Thus, we immediately obtain $g^{(k)} = (g_I^{(k)}, \tilde{u}_\Gamma^{(k)})$ and

$$g_{I,l+1}^{(k)} := g_{I,l}^{(k)} - \left(D\tilde{K}(g_{I,l}^{(k)}, \tilde{u}_\Gamma^{(k)})_{II} \right)^{-1} \left(\tilde{K}_I(g_{I,l}^{(k)}, \tilde{u}_\Gamma^{(k)}) - \tilde{f}_I \right)$$

from (12) and (14), converging to $g_I^{(k)}$. Therefore, the linearized FETI-DP system writes as

$$(26) \quad \begin{bmatrix} D\tilde{K}(g^{(k)})_{II} & D\tilde{K}(g^{(k)})_{I\Gamma} & 0 \\ D\tilde{K}(g^{(k)})_{I\Gamma} & D\tilde{K}(g^{(k)})_{\Gamma\Gamma} & B_\Gamma^T \\ 0 & B_\Gamma & 0 \end{bmatrix} \begin{bmatrix} \delta\tilde{u}_I^{(k)} \\ \delta\tilde{u}_\Gamma^{(k)} \\ \delta\lambda^{(k)} \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{K}_\Gamma(g^{(k)}) + B_\Gamma^T \lambda^{(k)} - \tilde{f}_\Gamma \\ Bg^{(k)} \end{bmatrix}.$$

As in NL-3, local nonlinear problems have to be solved in each Newton step of NL-4, but in NL-4 we only eliminate the interior variables \tilde{u}_I . Thus, the local solves are cheaper compared to NL-3. As in NL-3, in the inner Newton iteration of NL-4 no coarse problem is solved.

In addition to NL-3 and NL-4, other choices of E and L are possible. This includes problem-adaptive choices, e.g., based on the norm of the local nonlinear residuals on the interface or based on adaptive coarse space strategies for linear FETI-DP and BDDC.

2.5.9. Remarks on the nonlinear preconditioners. We provide a brief discussion on the properties of the different nonlinear preconditioners M . All four preconditioners obviously satisfy properties 1 and 2 from Figure 2 by definition of M . Also, property 3 holds, since omitting the constraint $Bu = 0$ makes M clearly easier to compute than A ; compare the results for NK_outer versus NL-2_inner, NL-3_inner, and NL-4_inner in Figure 13.

The computationally cheapest preconditioner is, of course, the identity (see NL-1), but it clearly does not give a good approximation of A . To compensate, a special initial value $\tilde{u}^{(0)}$ is computed from the nonlinear problem $\tilde{K}(\tilde{u}^{(0)}) = \tilde{f} - B^T \lambda^{(0)}$ by a Newton-type iteration, where $\lambda^{(0)}$ is some given initial value, often chosen as zero; see, e.g., [28]. The computational cost of this specific initial value is comparable to one application of the preconditioner M in NL-2, which is the most expensive preconditioner. This preconditioner also includes a nonlinear coarse problem. If a good coarse space is chosen, M will be a good nonlinear preconditioner of A ; i.e., M will be a good approximation of the inverse of A . In the two remaining variants NL-3 and NL-4, the computation of M only includes local solves and is thus computationally cheap and embarrassingly parallel. However, M will only be a good nonlinear preconditioner for A in NL-4 if the nonlinearities of the problem are strictly local to the subdomains; cf. our numerical results in subsection 5.6.

2.6. Using algebraic multigrid (AMG) for the coarse problem of nonlinear FETI-DP methods. If an inexact (e.g., multilevel) solver is used for the coarse problem of standard (linear or nonlinear) FETI-DP methods, then the solution is perturbed because the coarse problem is part of the operator and not part of the preconditioner. Therefore, a different approach has been taken for (linear or nonlinear) FETI-DP methods on very large scale parallel computers [31, 38].

In the outermost Newton iteration of all four of our nonlinear FETI-DP methods a linear system of the form

$$\begin{bmatrix} (D\tilde{K}(\tilde{u}))_{BB} & (D\tilde{K}(\tilde{u}))_{\Pi B}^T & B_B^T \\ (D\tilde{K}(\tilde{u}))_{\Pi B} & (D\tilde{K}(\tilde{u}))_{\Pi\Pi} & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{u}_B \\ \tilde{u}_\Pi \\ \lambda \end{bmatrix} = rhs_1$$

is solved; see (17) and also (20), (23), (25), and (26). Here, rhs_1 denotes the right-hand side, which differs for the different nonlinear methods. In order to construct FETI-DP methods which allow us to replace the exact factorization of the coarse problem $\tilde{S}_{\Pi\Pi}$ (see (19)) by a single iteration of a multilevel preconditioner, one step of linear block elimination of \tilde{u}_B is performed, resulting in

$$\begin{bmatrix} \tilde{S}_{\Pi\Pi} & -(D\tilde{K}(\tilde{u}))_{\Pi B} (D\tilde{K}(\tilde{u}))_{BB}^{-1} B_B^T \\ -B_B (D\tilde{K}(\tilde{u}))_{BB}^{-1} (D\tilde{K}(\tilde{u}))_{\Pi B}^T & -B_B (D\tilde{K}(\tilde{u}))_{BB}^{-1} B_B^T \end{bmatrix} \begin{bmatrix} \tilde{u}_\Pi \\ \lambda \end{bmatrix} = rhs_2;$$

cf. [36], where this idea was introduced. Now, a block triangular preconditioner for

1. $M : \widehat{W} \rightarrow \widehat{W}$,
2. M puts the current iterate into the neighborhood of the solution (see also [7]),
and
3. $M(x)$ is easily computable compared to the inverse action of $A(x)$.

FIG. 4. *Properties of the nonlinear preconditioner M .*

saddle point systems is used (see [36, 31, 38]) in combination with GMRES. Alternatively, the well-known symmetric positive definite reformulation can be used with conjugate gradients [6, 26, 36]. In our numerical experiments in section 5, we use one iteration of BoomerAMG as a preconditioner for the $\tilde{S}_{\Pi\Pi}$ block in the block triangular preconditioner and apply GMRES as a Krylov method. The lower right block is preconditioned using the standard FETI-DP Dirichlet preconditioner [19, 52] as in [36].

Using BoomerAMG [23] provides substantial leverage for the scalability of FETI-DP methods: using special interpolations, BoomerAMG is parallel scalable for linear elasticity problems for more than half a million cores [2].

3. Nonlinear BDDC framework. We can also consider nonlinear BDDC methods, introduced in [28], but we have to apply some generalizations to our framework. Solving a nonlinearly right-preconditioned problem

$$A(M(x)) = 0$$

with a Newton–Krylov method and a linear preconditioner for the Krylov method can be written as in Figure 5. The nonlinear FETI-DP methods can again be obtained by defining $x := (\tilde{u}, \lambda)$, the nonlinear function A as in (1), and M as before. Additionally, we can now derive a nonlinear BDDC method. Since BDDC methods operate on the assembled system, we now define

$$(27) \quad A(x) = A(\bar{u}) := R^T K(R\bar{u}) - R^T f.$$

Here, $K(R\bar{u}) - f = 0$ contains the local and decoupled problems on the subdomains. The variables in $\bar{u} \in \widehat{W}$ are assembled on the interface, and R is the restriction from the assembled (global) variables to the local subdomain variables. Thus, R^T operates as an assembly operator. We also define the restriction from the global interface to the local interfaces by R_Γ . A nonlinear right-preconditioner M for the assembled system has to satisfy the properties given in Figure 4. With all these definitions Figure 5 defines the nonlinear BDDC approach.

3.1. Nonlinear BDDC. With Figure 5 in mind, we can derive the nonlinear BDDC method introduced in [28] by defining the corresponding preconditioner M . As in NL-4, we decompose all degrees of freedom into interior (I) and interface (Γ) and obtain

$$A(\bar{u}) = \begin{pmatrix} A_I(R\bar{u}) \\ A_\Gamma(R\bar{u}) \end{pmatrix} = \begin{pmatrix} K_I(R\bar{u}) - f_I \\ R_\Gamma^T K_\Gamma(R\bar{u}) - R_\Gamma^T f_\Gamma \end{pmatrix} = R^T K(R\bar{u}) - R^T f.$$

We also split $\bar{u} = (\bar{u}_I, \bar{u}_\Gamma)$ and define the nonlinear preconditioner M by

$$M(\bar{u}) := (M_I(\bar{u}), \bar{u}_\Gamma),$$

```

Init:  $x^{(0)}$ 
Iterate over  $k$ :
  Compute:  $g^{(k)} := M(x^{(k)})$ 

  If  $\|A(g^{(k)})\|$  sufficiently small
    break; /* Convergence of nonlinear right-preconditioned method */

  Solve iteratively with some preconditioner:
   $DA(g^{(k)})DM(x^{(k)})\delta x^{(k)} = A(g^{(k)})$ 
  Update:  $x^{(k+1)} := x^{(k)} - \alpha^{(k)}\delta x^{(k)}$ 
End Iteration
    
```

FIG. 5. Generalized nonlinear algorithm.

where $M_I(\bar{u})$ is the solution of

$$(28) \quad K_I(M_I(\bar{u}), R_\Gamma \bar{u}_\Gamma) - f_I = 0.$$

Using

$$D(A(M(\bar{u}))) = \begin{pmatrix} DK_{II}(RM(\bar{u})) & DK_{I\Gamma}(RM(\bar{u}))R_\Gamma \\ R_\Gamma^T DK_{\Gamma I}(RM(\bar{u})) & R_\Gamma^T DK_{\Gamma\Gamma}(RM(\bar{u}))R_\Gamma \end{pmatrix},$$

$$DM(\bar{u}) = \begin{pmatrix} 0 & -DK_{II}^{-1}(\bar{u})DK_{I\Gamma}R_\Gamma \\ 0 & I \end{pmatrix},$$

and

$$A(M(\bar{u})) = \begin{pmatrix} 0 \\ R_\Gamma^T K_\Gamma(RM(\bar{u})) - R_\Gamma^T f_\Gamma \end{pmatrix},$$

we obtain the nonlinear BDDC method suggested in [28]. As in nonlinear FETI-DP methods, we can remove the inner derivative $DM(\cdot)$ from the tangential system without changing the update $\delta \bar{u}_\Gamma^{(k)}$ of the interface variables, where $\delta x^{(k)} = (\delta \bar{u}_I^{(k)}, \bar{u}_\Gamma^{(k)})$. The current iterate in the inner variables $\delta \bar{u}_I^{(k)}$ again only serves as initial value for the computation of $g^{(k+1)}$.

For numerical results for nonlinear BDDC, we refer the reader to [28, section 5.3].

3.2. Local convergence analysis for nonlinear BDDC. The necessary assumptions and resulting convergence properties are similar to those discussed in subsection 2.3 for nonlinear FETI-DP methods.

ASSUMPTION 3.1. *Let U_{BDDC} be an open neighborhood of our solution \bar{u}^* . We assume that $A(\bar{u})$ is continuously differentiable in U_{BDDC} and that $DA(\bar{u}^*)$ is a nonsingular matrix.*

If Assumption 3.1 is satisfied, Newton’s method solving $A(\bar{u}) = 0$ will converge for all initial values $\bar{u}^{(0)} \in U_{\text{BDDC}}^* \subset U_{\text{BDDC}}$ to the solution \bar{u}^* ; see, e.g., Ortega and Rheinboldt [49, section 10.2.2].

ASSUMPTION 3.2. *Let V_{BDDC}^* be an open neighborhood of \bar{u}^* . The function evaluation $M(\bar{u})$ is well defined and computable in V_{BDDC}^* and $M(V_{\text{BDDC}}^*) \subset U_{\text{BDDC}}^*$, with U_{BDDC}^* as defined above.*

Under Assumption 3.2, we have for all iterates $\bar{u}^{(k)} \in V_{\text{BDDC}}^*$ that $M(\bar{u}^{(k)}) \in U_{\text{BDDC}}^*$, and thus, with Assumption 3.1, the nonlinear BDDC method converges for all initial values $\bar{u}^{(0)} \in V_{\text{BDDC}}^*$ to the solution \bar{u}^* .

<pre> $g_0^{(k)} = x^{(k)}$ and $l = 0$ while $\ A_E(g_l^{(k)})\ > \epsilon_I$ Update with Newton's method to $g_{l+1}^{(k)}$ $l = l + 1$ $g^{(k)} = g_l^{(k)}$ end </pre>	<pre> $g_0^{(k)} = x^{(k)}$ and $l = 0$ $J_{old} = \frac{1}{2}\ A(g_0^{(k)})\ ^2$ while $\ A_E(g_l^{(k)})\ > \epsilon_I$ Update with Newton's method to $g_{l+1}^{(k)}$ Compute $J_{new} = \frac{1}{2}\ A(g_{l+1}^{(k)})\ ^2$ if $J_{new} > \tau J_{old}$ $g^{(k)} = g_l^{(k)}$ break while else $e_{old} = e_{new}$ end $l = l + 1$ $g^{(k)} = g_l^{(k)}$ end </pre>
--	--

FIG. 6. *Left: Computation of $M(x^{(k)})$. Right: Computation of $\mathcal{M}(x^{(k)})$.*

4. Controlling the inner Newton iteration. Several of our methods exhibit an inner and outer Newton iteration. In this section, we consider a strategy for choosing the accuracy of the inner Newton iteration in these methods. It is based on testing the reduction of the outer energy after each inner Newton step. We thus ensure that the local nonlinear elimination process will not interfere with the outer energy descent steps. We will see that this strategy enhances robustness and reliability and can increase the convergence radius significantly. Some additional details can also be found in [34].

It is a common globalization strategy for Newton-type methods to enforce a reduction of an energy, e.g., $J(x) = \frac{1}{2}\|A(x)\|^2$ in each iteration. If a condition on the Newton direction is not met, then the Newton step is rejected and replaced, e.g., by a gradient step. The step length is often controlled by a line search approach fulfilling certain conditions, such as, e.g., the Armijo or Wolfe condition; see [48]. If such descent strategies are applied for a nonlinear energy J , it seems desirable that the application of a nonlinear preconditioner M after a descent step should not increase the energy.

In our nonlinear FETI-DP and BDDC methods with an inner and an outer Newton iteration, the application of the nonlinear preconditioner M corresponds to a local nonlinear elimination step performed by a local Newton iteration.

Using our notation from section 3, we can write the nonlinear elimination performed by M (see (8) for FETI-DP and (28) for BDDC) as

$$A_E(g^{(k)}) = 0,$$

where $g^{(k)} = M(x^{(k)})$; see also Figure 6 (left). Therefore, in Figure 6 (left), we strive to minimize the energy $\frac{1}{2}\|A_E(x)\|^2$, but, unfortunately, we have no guarantee that we will also obtain a reduction of the energy $J(x) = \frac{1}{2}\|A(x)\|^2$. Our idea is therefore to stop the inner Newton iteration whenever a descent in the global energy corresponding to $J(g_{l+1}^{(k)}) \leq \tau J(g_l^{(k)})$ is not achieved and then set $\mathcal{M}(x^{(k)}) := g_l^{(k)}$; see also Figure 6 (right). Throughout this paper $0 < \tau \leq 1$ is set to 0.8.

This strategy corresponds to approximating $M(x)$ by an $\mathcal{M}(x)$ which does not

increase the energy J . Clearly, the approach never leads to an increased number of inner Newton iterations. In an extreme case, already the first inner Newton step can be rejected and $\mathcal{M}(x)$ reduces to the identity. In the context of nonlinear FETI-DP methods, the latter case is identical to a step of the NL-1 method.

Let us remark that in the very first outer iteration we at least spend two iterations in the computation of $\mathcal{M}(x^{(0)})$ to improve the initial value $x^{(0)}$. Let us further remark that, caused by the approximate solution of (8) or (28), the property (15) for FETI-DP (and the corresponding property for BDDC) will generally not hold; i.e., $DA \cdot DM$ will no longer be identical to the derivative of the nonlinear Schur complement. Also, the first entry of the right-hand sides in (16) and (17) can no longer be assumed to be zero. Therefore, we switch to an approximate tangential system and solve the linear system

$$DA(\mathcal{M}(x^{(k)}))\delta x^{(k)} = A(\mathcal{M}(x^{(k)}))$$

in each outer iteration. Nonetheless, at convergence all mentioned linear systems will be identical.

We refer to nonlinear FETI-DP and BDDC methods using \mathcal{M} as NL-ane-FETI-DP (approximate nonlinear elimination FETI-DP) and NL-ane-BDDC. The different nonlinear FETI-DP variants are denoted by NL-ane-2, NL-ane-3, and NL-ane-4. A similar approach can be applied to the initial value computation often used in NL-1, but we do not consider this variant here.

5. Numerical results.

5.1. Nonlinear model problems. As a model problem, we define the scaled p -Laplace operator for $p \geq 2$ by

$$\alpha \Delta_p u := \operatorname{div}(\alpha |\nabla u|^{p-2} \nabla u).$$

We then consider

$$\begin{aligned} -\alpha \Delta_p u - \beta \Delta_2 u &= 1 && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where $\alpha, \beta : \Omega \rightarrow \mathbb{R}$ are coefficient functions.

We would like to consider two types of problems—first, one with nonlinearities completely contained in the interior of subdomains, and second, a problem where nonlinearities have a global character.

For the problem type *Localized Nonlinearities*, we consider subdomains with inclusions defined by

$$\alpha(x) = \begin{cases} 1 & \text{if } x \in \Omega_I, \\ 0 & \text{elsewhere,} \end{cases} \quad \beta(x) = \begin{cases} 0 & \text{if } x \in \Omega_I, \\ 1 & \text{elsewhere;} \end{cases}$$

see Figure 7 (left) for a possible shape of the localized nonlinearities (nonlinear inclusions) Ω_I in two dimensions. For *Localized Nonlinearities* in two dimensions, we distinguish between the problem *Standard Inclusions*, where we exclusively consider square subdomains with square inclusions, see Figure 7 (left), and the problem *Nonstandard Inclusions*, where the inclusions can have the shape of a rectangle, a cross, or an ellipse (approximated on a regular grid); see Figure 8 (lower). For *Standard Inclusions* we consider different rectangles and the unit square as computational domain Ω , and for *Nonstandard Inclusions* we exclusively consider a curved domain; see Figure 8

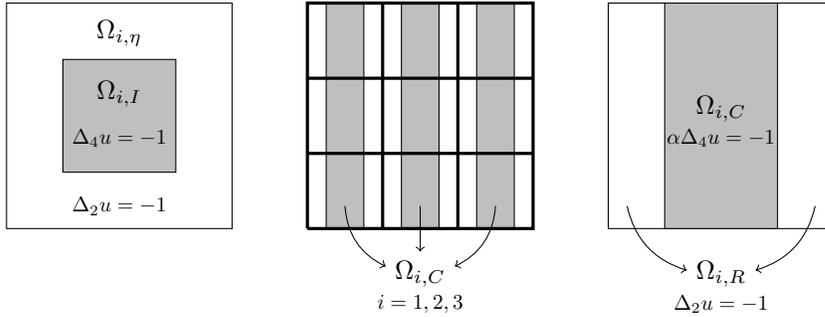


FIG. 7. *Left (localized nonlinearities):* Subdomain Ω_i with an inclusion $\Omega_{i,I}$ surrounded by a hull $\Omega_{i,\eta}$ with width η . Define $\Omega_I := \bigcup_i \Omega_{i,I}$. *Middle (nonlocal nonlinearities):* Example for a decomposition of Ω in $N = 9$ subdomains, intersected by 3 channels $\Omega_{i,C}, i = 1, 2, 3$. We define $\Omega_C = \bigcup_i \Omega_{i,C}$. *Right:* Subdomain Ω_i with channel $\Omega_{i,C}$ of width $\frac{H}{2}$.

(lower). In three dimensions, we consider a cuboid domain Ω with centered spherical inclusions (approximated on a regular grid).

For the problem type *Nonlocal Nonlinearities*, we consider two different arrangements of nonlinearities. The first problem is called *Channels*, where we consider subdomains with channels defined by

$$\alpha(x) = \begin{cases} 1e5 & \text{if } x \in \Omega_C, \\ 0 & \text{elsewhere,} \end{cases} \quad \beta(x) = \begin{cases} 0 & \text{if } x \in \Omega_C, \\ 1 & \text{elsewhere,} \end{cases}$$

where the channels Ω_C are depicted in Figure 7 (middle and right) for the case of two dimensions. The second problem is called *Grid*. Here, the nonlinearities have a more global character compared to the *Channels* problem. The subdomains are intersected by a grid of channels, which does not touch the boundary of the domain. It is defined by

$$\alpha(x) = \begin{cases} 1 & \text{if } x \in \Omega_G, \\ 0 & \text{elsewhere,} \end{cases} \quad \beta(x) = \begin{cases} 0 & \text{if } x \in \Omega_G, \\ 1 & \text{elsewhere,} \end{cases}$$

where the grid Ω_G is depicted in Figure 8 (left) for the case of two dimensions.

5.2. General remarks. In all our tables, we refer to the traditional Newton–Krylov–FETI-DP method as the NK method and to the nonlinear FETI-DP variants as before as NL- i , $i = 1, \dots, 4$ (see subsection 2.4), and NL-ane- k , $k = 2, \dots, 4$, respectively (see section 4). For simplicity, we use linear finite elements, and we always use a coarse space constructed from all primal vertices only. Of course, stronger coarse spaces can also be used with nonlinear FETI-DP methods; see [28] for the effects of different coarse spaces enforced by a transformation of basis approach in nonlinear FETI-DP.

For all methods, we compare the execution time (exec. time). This includes the time to assemble and to solve the problem. The lowest numbers are marked in bold. To provide a fair comparison, we always compute the parallel efficiency using as a base line the fastest of the five approaches on the smallest number of processor cores considered. That implies parallel efficiencies below 100% for four of the five methods already for the smallest computations.

To allow for a better interpretation of the computing times, for each method, we provide the number of necessary factorizations of the FETI-DP coarse problem

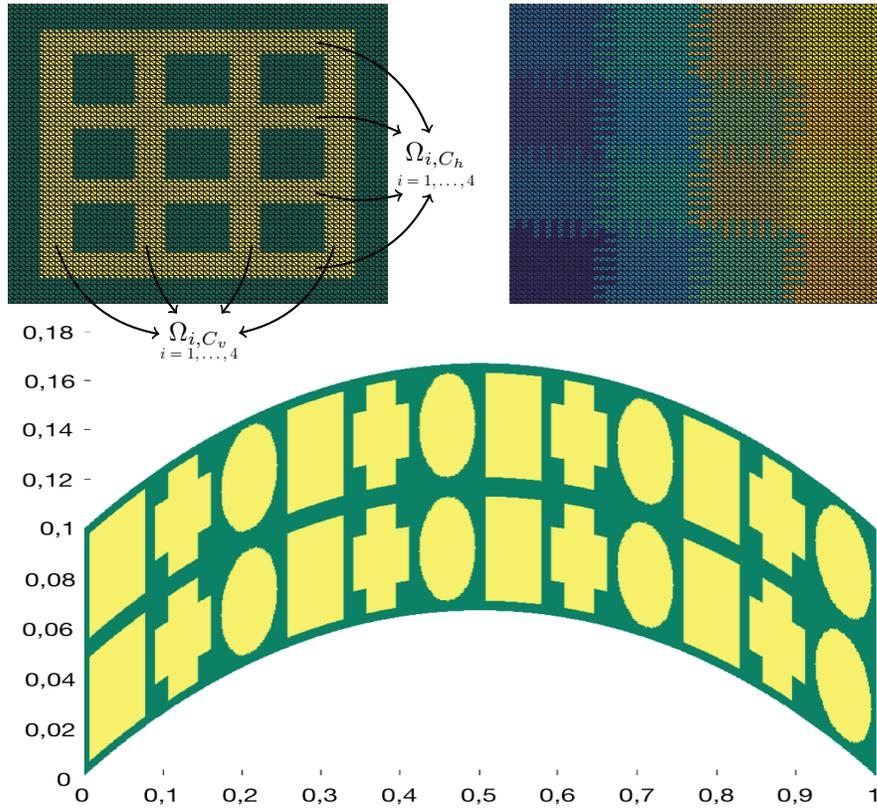


FIG. 8. Left: p -Laplace in yellow grid and Laplace in matrix material. Define $\Omega_G := \bigcup_i \Omega_{i,C_h} \cup \Omega_{i,C_v}$. Right: Domain Decomposition with ragged edges. Bottom: p -Laplace in the Nonstandard Inclusions and Laplace in matrix material for a curved domain.

\tilde{S}_{III} (denoted by “Coarse Factor.”), the number of local factorizations of $D\tilde{K}_{BB}$ or $D\tilde{K}_{II}$ (denoted by “Local Factor.”), and the sum of the Krylov iterations over all Newton steps (denoted by “Krylov It.”) in the tables; see also [28, 31]. For NL-1 and NL-2 coarse factorizations are not only necessary in the main loop but also in the computation of the initial value in NL-1 and in the inner loop in NL-2. Therefore, we subdivide the number of necessary factorizations of \tilde{S}_{III} into coarse factorizations in the outer loop (denoted by “out.”) and coarse factorizations in the computation of the initial value or the inner loop (denoted by “in.”), respectively. For all nonlinear methods the number of coarse factorizations in the main loop is equivalent to the number of outer Newton steps.

The number of subdomains is identical to the number of message passing interface (MPI) ranks in all our experiments. We formulate identical stopping criteria for all our FETI-DP algorithms based on \tilde{u} : The global Newton iteration will be stopped if the fully assembled nonlinear residual is smaller than ε_O . Newton iterations resulting from systems of the form $\tilde{K}(M_u(\tilde{u}, \lambda)) + B^T \lambda - \tilde{f} = 0$ (see (8) and (21)) will be stopped if $\|\tilde{K}(M_u(\tilde{u}, \lambda)) + B^T \lambda - \tilde{f}\|_{L_2}$ is smaller than the minimum of ε_I and the norm of the fully assembled residual in the current outer iteration multiplied by $1e-2$. Using such a stopping criterion, we avoid unnecessary exactness in early outer Newton iterations and still obtain a sufficiently exact solution at convergence. The values of

$\varepsilon_O \in \{1e-8, 1e-12\}$ and $\varepsilon_I \in \{1e-5, 1e-6, 1e-7\}$ vary over different computations; see the captions of the tables.

If not noted otherwise, we solve the linearized systems using the preconditioned conjugate gradient method (PCG) as a Krylov space method. The tangential matrix of the p -Laplacian is always symmetric positive definite if not evaluated in constant functions. Since we always have zero Dirichlet boundary conditions and use a nonzero initial value, we avoid the latter case and can safely use the PCG method. In case of inexact reduced FETI-DP-type methods (see subsection 2.6), we choose GMRES as a Krylov method, since the block-triangular preconditioner is not symmetric, and we do not use the standard symmetric positive definite reformulation; cf. [6, 36]. For all Krylov iterations, we use a relative residual tolerance of $1e-10$, which might be overly exact, especially in the first Newton steps. More advanced techniques to choose the forcing terms in inexact Newton's methods can be found in, e.g., [16], but this is not the focus of this paper.

5.3. Computational platforms and implementation. We perform our computations on a Tier-3, Tier-2, and a Tier-1/Tier-0 supercomputer of the German High Performance Computing Pyramid:

- JUQUEEN (Tier-1/0): 458 752 Blue Gene/Q cores (PowerPC A2 1.6 GHz; 16 cores and 16 GB per node); 5.9 PFlops; operated by Jülich Supercomputing Center (JSC) providing computing time for Germany and Europe; ranked 19th in the current TOP500 list (November, 2016).
- Taurus (Tier-2): 34 656 Xeon cores (2 020 nodes); 1.4 PFlops; operated by Center for Information Services and High Performance Computing (ZIH) of the TU Dresden providing HPC resources for Saxony; TOP500 rank 107 (November, 2016).
- MagnitUDE (Tier-3): 13 536 cores (Broadwell XEON E5-2650v4 12C 2.2GHz; 24 cores and 72 GB per node); 476.5 TFlops NEC Cluster; operated by Center for Computational Sciences and Simulation (CCSS) of the Universität Duisburg-Essen (UDE) providing computing resources for UDE; TOP500 rank 384 (November, 2016).

On Taurus, we use a Haswell XEON E5-2680v3 12C 2.5GHz processor partition with 24 cores and 64 GB memory per node.

We have implemented all nonlinear FETI-DP variants as well as the Newton–Krylov–FETI-DP method in PETSc [3, 4] using a common software framework, i.e., making use of the same software building blocks. Thus, our comparison in terms of runtime is fair. Note that we do not make use of the built-in BDDC implementation of PETSc. For all local sparse factorizations, we use the latest version of UMFPACK [13]. The publication [31] provides details on the implementation which we build on.

5.4. Localized nonlinearities in two dimensions.

5.4.1. Standard exact FETI-DP methods. We first discuss standard linear and nonlinear FETI-DP methods, i.e., using exact (sparse) solvers for the subdomains and the coarse problem. We consider the model problem *Localized Nonlinearities* with $p = 4$; see section 5. As a domain Ω we consider a rectangle $(0, 1.5) \times (0, 1)$ (see Tables 1 and 2) or the curved domain (see Table 3). As mentioned in subsection 5.1, we consider *Standard Inclusions* in combination with the rectangle and *Nonstandard Inclusions* in combination with the curved domain; see Figure 8 (lower) for the arrangement of inclusions for 24 subdomains. For the nonlinear FETI-DP methods NL-2, NL-3, and NL-4, the nonlinearities are completely contained in the index set E . As a result, we

expect the inner Newton iterations in these methods to be effective in reducing the number of outer Newton iterations. For the NL-1 method, which only has an outer Newton iteration, the computation of the initial value takes the role of the inner Newton iteration.

We consider subdomains with 160K degrees of freedom ($H/h = 400$) on magnitUDE (see Tables 1 and 3) and Taurus (see Table 2) using very similar settings; see the captions of the tables. On magnitUDE, we use two MPI ranks per core (making use of the hyperthreads for MPI processes) since the supercomputer has only 13 536 cores. We indeed observed, for our application, a modest performance gain of about 10 percent from using two MPI ranks per core on magnitUDE over the use of a single MPI rank per core. We do not use threading. On Taurus, we use 24 576 MPI ranks and cores.

In our experiments in Table 1, Table 2, and Table 3, NL-2 is the fastest method and, especially, always faster than standard NK (Newton–Krylov) method. While NL-4 is always slower than NK for a small number of MPI ranks, NK is the slowest method beyond 96 ranks for *Standard Inclusions*. For *Nonstandard Inclusions* on the curved domain NK and NL-4 have identical execution times for the largest test. In fact, for large problems, the nonlinear FETI-DP methods NL-2, NL-3, and NL-4 are about twice as fast compared to the traditional NK approach. As expected, this is a result of a significantly reduced number of Krylov iterations (see “Krylov It.”) for the nonlinear FETI-DP approaches compared to NK. This, in turn, is achieved by investing more local work; see the number of local factorizations.

The NL-2 method has the most effective inner Newton iteration; i.e., the inner iteration includes local elimination and global transport of information since weakly coupled nonlinear problems are solved. Indeed, it achieves the largest reduction in the number of Krylov iterations and the fastest computing times. In contrast, in the NL-3 and NL-4 methods no coarse problem has to be solved in the inner Newton iteration. However, for the number of cores available on these Tier-2 and Tier-3 supercomputers, the cost for the coarse problem does not seem to be significant enough; i.e., the savings in the number of the sum of coarse solves (see “Coarse Factor.”) cannot compensate for the higher number of local factorizations and Krylov iterations observed for NL-3 and NL-4.

The NL-1 method always gives results falling in between results of the NK method and the other nonlinear methods. This is not surprising since, except for the computation of the initial value, it is algorithmically closely related to the NK method.

Note that, for 24 576 ranks, for both supercomputers, we see a significant drop in parallel efficiency; e.g., for the NL-2 method the parallel efficiency decreases from above 70 percent to below 50 percent. To obtain better scalability for a large number of cores and ranks, in the next section, we therefore will switch to a multilevel solver for the FETI-DP coarse problem, using the approach from subsection 2.6.

5.4.2. Scalability on a Tier-0 supercomputer. We now present results for the European Tier-0 supercomputer JUQUEEN. The JUQUEEN supercomputer also provides the German national Tier-1 level.

First, we use exact solvers for the subdomain problems and the coarse problem. We use smaller subdomains to accommodate for the slower BlueGene/Q PowerPC cores and the smaller amount of memory per core compared to the other x86-based supercomputers. To make efficient use of the hardware threads of the Blue Gene/Q processor, it is advisable to use threading or multiple MPI ranks per core since significant performance gains of almost a factor of two can be achieved [29]. We always use

TABLE 1

Model problem *Localized Nonlinearities – Standard Inclusions* (see subsection 5.1). Nonlinear FETI-DP algorithms compared to the more traditional Newton–Krylov–FETI-DP; domain $\Omega = (0, 1.5) \times (0, 1)$ decomposed into square subdomains; $p = 4$; $H/h = 400$; $\eta = 20h$; $\varepsilon_I = 1e - 7$; $\varepsilon_O = 1e - 12$; two MPI ranks per core; computed on magnitUDE.

<i>Localized Nonlinearities - Standard Inclusions</i>								
2D; $p = 4$; $H/h = 400$; exact FETI-DP; computed on magnitUDE								
MPI ranks	Problem size	Nonlinear solver	Local factor.	Coarse factor.		Krylov it.	Execution time	Parallel eff.
				in.	out.			
24	3 844 001	NK	20	-	20	363	171.01s	63%
		NL-1	23	11	12	224	142.64s	76%
		NL-2	26	22	4	73	108.09s	100%
		NL-3	40	0	5	91	148.77s	73%
		NL-4	42	0	5	98	171.81s	63%
96	15 368 001	NK	19	-	19	499	191.08s	57%
		NL-1	25	12	13	345	167.56s	65%
		NL-2	27	23	4	105	119.05s	91%
		NL-3	43	0	5	132	166.60s	65%
		NL-4	37	0	5	144	164.66s	66%
384	61 456 001	NK	21	-	21	619	222.28s	49%
		NL-1	25	12	13	351	176.12s	61%
		NL-2	29	25	4	117	130.29s	83%
		NL-3	43	0	5	144	173.42s	62%
		NL-4	38	0	5	162	176.77s	61%
1 536	245 792 001	NK	24	-	24	738	265.48s	41%
		NL-1	33	12	21	541	250.05s	43%
		NL-2	30	26	4	120	136.43s	79%
		NL-3	43	0	5	150	175.94s	61%
		NL-4	41	0	5	168	190.05s	57%
6 144	983 104 001	NK	25	-	25	802	297.77s	36%
		NL-1	29	15	14	411	219.46s	49%
		NL-2	32	28	4	125	149.87s	72%
		NL-3	47	0	5	157	196.45s	55%
		NL-4	45	0	5	173	213.16s	51%
24 576	3 932 288 001	NK	26	-	26	871	485.9s	22%
		NL-1	29	15	14	400	313.19s	35%
		NL-2	35	31	4	127	225.31s	48%
		NL-3	47	0	5	159	235.17s	46%
		NL-4	43	0	5	177	240.28s	45%

two MPI ranks per core on JUQUEEN as, for our application, this makes the most efficient use of the hardware threads; see also our JUQUEEN results in [29, 31]. The results in Table 4 for up to 16 384 MPI ranks, using a square domain and *Standard Inclusions*, are comparable to the ones presented in subsection 5.4.1 for magnitUDE and Taurus.

As a result of the capable network of Blue Gene machines, for all methods, good weak parallel scalability is achieved. All nonlinear methods perform significantly better than the NK method; see also Table 4 and Figure 9.

An even better parallel scalability is prevented by the cost for exactly solving the FETI-DP coarse problem. To obtain better scalability for and beyond 32 768 MPI ranks, we apply an AMG preconditioner (see subsection 2.6) to the FETI-DP coarse problem instead of using a sparse direct solver by applying an inexact reduced FETI-DP approach. Using this approach, we obtain weak parallel scalability to 131 072 MPI ranks; see Table 5 and Figure 10.

Note that in the case of inexact reduced FETI-DP, additional Krylov iterations have to be performed to solve the FETI-DP coarse problem using the BoomerAMG preconditioner in the inner loop of NL-2 and the computation of the initial value of NL-1; see also [31] for a detailed discussion. These iterations are computationally cheaper than the Krylov iterations carried out in the outer loop and thus are counted

TABLE 2

Model problem *Localized Nonlinearities – Standard Inclusions* (see subsection 5.1). Nonlinear FETI-DP algorithms compared to the more traditional Newton–Krylov–FETI-DP; domain $\Omega = (0, 1.5) \times (0, 1)$ decomposed into square subdomains; $p = 4$; $H/h = 400$; $\eta = 10h$; $\varepsilon_I = 1e-6$; $\varepsilon_O = 1e-12$; one MPI rank per core; computed on Taurus.

Localized Nonlinearities - Standard Inclusions								
2D; $p = 4$; $H/h = 400$; exact FETI-DP; computed on Taurus								
MPI ranks	Problem size	Nonlinear solver	Local factor.	Coarse factor.		Krylov it.	Execution time	Parallel effc.
				in.	out.			
24	3 844 001	NK	22	-	22	398	168.82s	71%
		NL-1	27	14	13	256	152.76s	79%
		NL-2	30	26	4	76	120.10s	100%
		NL-3	44	0	5	95	163.01s	74%
		NL-4	49	0	6	119	202.64s	59%
96	15 368 001	NK	24	-	24	636	209.14s	57%
		NL-1	27	13	14	406	169.67s	71%
		NL-2	31	27	4	111	127.64s	94%
		NL-3	48	0	5	137	182.16s	66%
		NL-4	50	0	6	176	211.86s	57%
384	61 456 001	NK	25	-	25	748	228.04s	53%
		NL-1	26	13	13	418	168.70s	71%
		NL-2	32	28	4	127	133.34s	90%
		NL-3	50	0	5	154	189.63s	64%
		NL-4	50	0	6	200	218.44s	55%
1 536	245 792 001	NK	24	-	24	781	282.94s	42%
		NL-1	28	14	14	472	219.81s	55%
		NL-2	35	31	4	132	161.50s	75%
		NL-3	48	0	5	161	197.70s	61%
		NL-4	50	0	6	210	238.76s	50%
6 144	983 104 001	NK	25	-	25	858	288.35s	42%
		NL-1	29	15	14	479	218.34s	55%
		NL-2	36	32	4	137	163.43s	74%
		NL-3	52	0	5	166	213.98s	56%
		NL-4	59	0	7	253	280.69s	43%
24 576	3 932 288 001	NK	27	-	27	975	598.40s	20%
		NL-1	29	15	14	485	426.03s	28%
		NL-2	38	34	4	140	348.10s	35%
		NL-3	51	0	5	169	386.57s	31%
		NL-4	53	0	6	223	436.51s	28%

separately and denoted by iterations in \tilde{S}_{III} in Table 5. Let us further remark that we usually do not distribute the coarse problem to all available MPI ranks but, e.g., only to 2% of the 131 072 MPI ranks for the largest computation; details can again be found in [31].

The results in Table 5 for the JUQUEEN supercomputer are qualitatively similar to the results discussed previously. However, for a larger number of ranks, i.e., for 8 192, 32 768, and 131 072 ranks, the NL-3 method is the fastest algorithm as it combines a cheap inner Newton iteration (without the need to solve a coarse problem) with a low number of outer Newton iterations. Although the NL-4 method is very similar to the NL-3 method, it is slower due to a significantly larger number of Krylov iterations in the outer loop.

5.5. Localized nonlinearities in three dimensions. We now consider the model problem *Localized Nonlinearities* in three dimensions with $p = 4$ and $H/h = 30$; see Tables 6 and 7. The subdomain-centered spherical inclusions have a diameter $0.6H$. The stopping criterion is not based on the norm of the nonlinear residual but on the norm of the update $\delta\tilde{u}$.

Again, all nonlinear FETI-DP methods reduce the number of Krylov iterations, increase the local work, and are faster compared to NK. For 24 576 MPI ranks, the fastest nonlinear methods (NL-2 and NL-3) are more than twice as fast as NK. As

TABLE 3

Model problem *Localized Nonlinearities – Nonstandard Inclusions* (see subsection 5.1). New nonlinear FETI-DP algorithms compared to the more traditional Newton–Krylov–FETI-DP; $p = 4$ and a weight of $\alpha = 1$ inside the inclusion and $p = 2$ and $\beta = 1$ elsewhere; domain Ω is a curved domain with a height of 0.1 and a width of 1.0; see also Figure 8 (bottom); decomposed into square subdomains; $H/h = 400$; $\varepsilon_I = 1e-5$; $\varepsilon_O = 1e-8$; the stopping criterion is based on the norm of $\delta\tilde{u}$; two MPI ranks per core; computed on magnitUDE.

<i>Localized Nonlinearities – Nonstandard Inclusions</i>								
2D; $p = 4$; $H/h = 400$; exact FETI-DP; computed on magnitUDE								
MPI ranks	Problem size	Nonlinear solver	Local factor.	Coarse factor.		Krylov it.	Execution time	Parallel effic.
				in.	out.			
24	3 844 001	NK	19	-	19	343	88.27s	60%
		NL-1	20	11	9	138	60.92s	87%
		NL-2	23	19	4	62	53.20s	100%
		NL-3	40	0	6	92	84.67s	63%
		NL-4	54	0	8	128	125.39s	42%
96	15 368 001	NK	21	-	21	568	116.85s	46%
		NL-1	27	9	18	350	107.51s	49%
		NL-2	31	26	5	107	75.58s	70%
		NL-3	41	0	6	142	91.81s	58%
		NL-4	50	0	9	229	135.84s	39%
384	61 456 001	NK	22	-	22	614	125.49s	42%
		NL-1	26	10	16	332	101.88s	52%
		NL-2	27	23	4	95	67.11s	79%
		NL-3	33	0	6	150	79.95s	67%
		NL-4	44	0	9	243	127.28s	41%
1536	245 792 001	NK	25	-	25	729	152.78s	35%
		NL-1	27	8	19	380	116.05s	46%
		NL-2	32	27	5	111	81.75s	65%
		NL-3	37	0	6	155	89.88s	59%
		NL-4	43	0	8	246	126.53s	42%
6144	983 104 001	NK	20	-	20	610	136.54s	39%
		NL-1	28	8	20	378	127.45s	42%
		NL-2	29	25	4	98	77.69s	68%
		NL-3	36	0	6	157	92.28s	58%
		NL-4	45	0	8	252	136.83s	39%

in two dimensions, we observe the expected decrease of coarse solves for NL-3 and NL-4, but we already benefit thereof for a smaller number of MPI ranks. While for the setting in Table 6, NL-4 appears to perform best and outpaces the other four methods, NL-3 has the shortest runtime for the larger domain $\Omega = (0, 4) \times (0, 4) \times (0, 3)$ considered in Table 7. The first observation is related to the smaller average time per inner loop for NL-4 (see Figure 11), while the latter effect is related to the comparable small number of outer Newton iterations in NL-3 in Table 7.

5.6. Nonlocal nonlinearities in two dimensions. We now consider our model problem *Channels* (see subsection 5.1); see also Figure 7 (middle and right) for the geometry.

Here, the nonlinearities are no longer restricted to a single subdomain each, and, for this problem, the nonlinearity is not contained in the index set E for the NL-4 method. This is opposed to the NL-2 and NL-3 methods.

The results for a computation of our model problem on magnitUDE are presented in Table 8 and Figure 12. Our expectations regarding a bad performance of NL-4 are confirmed, and NL-4 performs even worse than NK. For the remaining nonlinear FETI-DP methods, we obtain qualitatively similar results compared to tests with localized nonlinearities; see Tables 1 and 2. Again, NL-2 is the fastest method, and NL-3 just slightly catches up. Thus, the lower local work still outweighs the higher number of coarse solves for NL-2, at least for this comparably small number of MPI ranks.

TABLE 4

Model problem *Localized Nonlinearities – Standard Inclusions* (see subsection 5.1). New nonlinear FETI-DP algorithms (NL-1,-2,-3, and -4) compared to the more traditional Newton–Krylov–FETI-DP; domain $\Omega = (0, 1) \times (0, 1)$ and $H/h = 200$ decomposed into square subdomains; $\eta = 10h$; $\varepsilon_O = 1e-12$; $\varepsilon_I = 1e-7$; two MPI ranks per core; computed on JUQUEEN.

Localized Nonlinearities - Standard Inclusions								
2D; $p = 4$; $H/h = 200$; exact FETI-DP; computed on JUQUEEN								
MPI ranks	Problem size	Nonlinear solver	Local factor.	Coarse factor.		Krylov it.	Execution time	Parallel effc.
				in.	out.			
64	2 563 201	NK	21	-	21	443	236.96s	56%
		NL-1	21	15	6	126	144.82s	91%
		NL-2	23	20	3	66	131.68s	100%
		NL-3	36	0	5	105	193.75s	68%
		NL-4	38	0	6	135	227.14s	58%
256	10 246 401	NK	22	-	22	559	261.73s	50%
		NL-1	22	15	7	180	160.37s	82%
		NL-2	23	20	3	79	133.47s	99%
		NL-3	37	0	5	127	201.16s	66%
		NL-4	39	0	6	166	237.00s	56%
1024	40 972 801	NK	24	-	24	660	294.08s	45%
		NL-1	26	16	10	241	200.32s	66%
		NL-2	29	25	4	103	171.35s	77%
		NL-3	38	0	5	134	207.26s	64%
		NL-4	39	0	6	177	239.97s	55%
4096	163 865 601	NK	26	-	26	770	336.14s	39%
		NL-1	26	16	10	248	209.08s	63%
		NL-2	29	25	4	107	181.57s	72%
		NL-3	39	0	5	139	215.60s	61%
		NL-4	38	0	6	185	239.30s	55%
16384	655 411 201	NK	27	-	27	823	403.87s	33%
		NL-1	28	19	9	216	250.36s	53%
		NL-2	31	27	4	110	230.21s	57%
		NL-3	41	0	5	141	239.12s	55%
		NL-4	41	0	6	188	268.49s	49%

It is remarkable that the fastest nonlinear FETI-DP methods (NL-2 and NL-3) are, both, more than five times faster than the traditional NK approach for the largest problem in Table 8. Here, even the NL-1 method is more than twice as fast as NK. Only the NL-4 method, with an index set E inappropriate for the problem, gives results similar to those of NK. This illustrates that the choice of the elimination index set E is critical.

5.7. Controlling the inner Newton iteration: Numerical results. For our *Grid* problem, we consider a domain decomposition with ragged edges; see Figure 8 (right). Here, we present results for nonlinear FETI-DP methods with the additional ability of controlling the accuracy of the inner Newton iteration (section 4) using a sequential MATLAB implementation. The results are presented in Table 9. For completeness, we also present the results of the corresponding traditional nonlinear FETI-DP methods (see the numbers in brackets).

It turns out that NL-2 and NL-3 do not lead to convergence for and beyond 16 subdomains due to an insufficient coarse space, but NL-ane-2 and NL-ane-3 do not suffer from this. Both methods perform quite similarly up to 16 subdomains, but for a larger number of subdomains the nonlinear preconditioner of NL-ane-2 gets more effective. Nevertheless, NL-ane-3 also ends up with smaller numbers of Newton steps and Krylov iterations compared to NK and the closely related NL-1 method without the computation of an initial value. For 256 subdomains NL-ane-2 saves more than 50% of Newton steps and more than 66% of Krylov iterations compared to traditional NK.

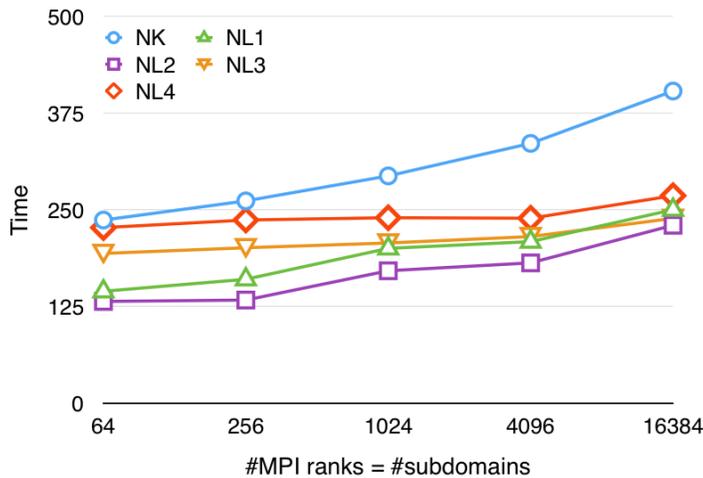


FIG. 9. Model problem Localized Nonlinearities – Standard Inclusions: Weak scalability of nonlinear FETI-DP algorithms (NL-1, -2, -3, and -4) and the more traditional Newton–Krylov–FETI-DP method (NK) on the JUQUEEN BlueGene/Q supercomputer at Forschungszentrum Jülich; data from Table 4.

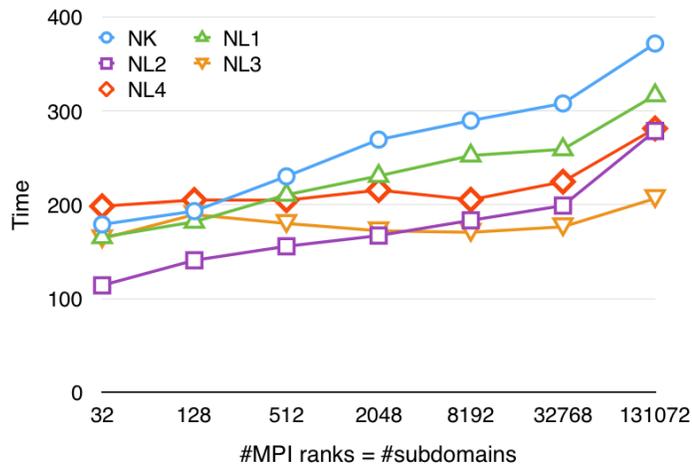


FIG. 10. Model problem Localized Nonlinearities – Standard Inclusions: Weak scalability of new inexact reduced FETI-DP algorithms and the inexact reduced version of the more traditional Newton–Krylov–FETI-DP algorithm on the JUQUEEN BlueGene/Q machine at Forschungszentrum Jülich; data from Table 5.

In contrast to NL-2 and NL-3, NL-4 converges to the correct solution for all numbers of subdomains, but the numbers of outer Newton steps and Krylov iterations are just slightly reduced compared to NL-1 without the computation of an initial value and NK, due to the fact that the nonlinearities are not contained in the index set E for NL-4. This, of course, also holds for NL-ane-4, and so the numbers of Newton steps and Krylov iterations of NL-4 and NL-ane-4 are quite similar. The big

TABLE 5

Model problem Localized Nonlinearities – Standard Inclusions (see subsection 5.1). Inexact reduced nonlinear FETI-DP algorithms compared to the inexact reduced version of the more traditional Newton–Krylov–FETI-DP. Here, we consider as a domain $\Omega = (0, 2) \times (0, 1)$ decomposed into square subdomains; $p = 4$; $H/h = 200$; $\eta = 10h$; $\varepsilon_I = 1e-6$; $\varepsilon_O = 1e-12$; two MPI ranks per core. Instead of the exact factorizations of \tilde{S}_{III} , we now have to set up an AMG preconditioner for \tilde{S}_{III} several times. We also have one AMG application per GMRES iteration; computed on the JUQUEEN supercomputer; also see Figure 10.

Localized Nonlinearities – Standard Inclusions									
2D; $p = 4$; $H/h = 200$; inexact reduced FETI-DP; computed on JUQUEEN									
MPI ranks	Problem size	Nonlin. solver	Local factor.	AMG setup		Krylov it.		Execution time	Parallel effc.
				in.	out.	\tilde{S}_{III}	Full		
32	1 282 401	NK	16	-	16	-	341	178.98s	64%
		NL-1	19	8	11	31	252	165.35s	69%
		NL-2	20	17	3	63	71	114.06s	100%
		NL-3	31	0	4	0	87	163.79s	70%
		NL-4	34	0	5	0	112	198.37s	58%
128	5 124 801	NK	16	-	16	-	419	193.17s	59%
		NL-1	20	9	11	41	319	181.79s	63%
		NL-2	23	19	4	82	123	140.77s	81%
		NL-3	34	0	5	0	137	189.46s	60%
		NL-4	34	0	5	0	141	205.06s	56%
512	20 489 601	NK	18	-	18	-	511	230.17s	50%
		NL-1	22	10	12	50	377	210.54s	54%
		NL-2	25	21	4	98	137	155.68s	73%
		NL-3	31	0	5	0	146	180.00s	63%
		NL-4	33	0	5	0	150	204.74s	56%
2 048	81 939 201	NK	21	-	21	-	646	269.30s	42%
		NL-1	24	11	13	55	444	230.64s	49%
		NL-2	27	23	4	106	150	167.10s	68%
		NL-3	31	0	4	0	122	172.22s	66%
		NL-4	35	0	5	0	168	215.56s	53%
8 192	327 718 401	NK	22	-	22	-	692	289.56s	39%
		NL-1	25	11	14	55	490	252.36s	45%
		NL-2	29	25	4	117	156	183.34s	62%
		NL-3	30	0	4	0	125	170.58s	67%
		NL-4	32	0	5	0	172	205.25s	51%
32 768	1 310 796 801	NK	23	-	23	-	722	307.87s	37%
		NL-1	26	12	14	60	472	259.02s	44%
		NL-2	31	27	4	134	155	199.05s	57%
		NL-3	30	0	4	0	121	176.47s	65%
		NL-4	35	0	5	0	165	224.48s	51%
131 072	5 243 033 601	NK	24	-	24	-	766	371.68s	31%
		NL-1	26	12	14	60	467	316.50s	36%
		NL-2	35	31	4	153	160	278.56s	41%
		NL-3	29	0	4	0	119	206.57s	55%
		NL-4	38	0	5	0	165	281.22s	41%

difference between methods NL-4 and NL-ane-4 is the number of local factorizations (inner Newton iterations), which is reduced by NL-ane-4 by more than 50%. Since the number of local factorizations of NL-ane-4 is less than two times the number of Newton steps, the elimination of u_I reduces the global energy J just in some cases. As a consequence, NL-ane-4 is quite close to NL-1 without computation of an initial value.

Controlling the accuracy of the inner Newton iteration cannot reduce outer Newton steps or Krylov iterations when the elimination set E is inappropriate for the problem, but it avoids unnecessary inner Newton steps. It also enlarges the convergence radius of nonlinear FETI-DP methods and reduces the dependency on the coarse space.

5.8. Better scalability in nonlinear methods from localizing work. The nonlinear FETI-DP methods achieve their better scalability by localizing computa-

TABLE 6

Model problem *Localized Nonlinearities in 3D* (see subsection 5.1). New nonlinear FETI-DP algorithms compared to the more traditional Newton–Krylov–FETI-DP; domain $\Omega = (0, 1.5) \times (0, 1) \times (0, 1)$ decomposed into cubic subdomains; $p = 4$; $H/h = 30$; centered spherical inclusions with diameter $0.6H$; $\varepsilon_I = 1e-5$; $\varepsilon_O = 1e-8$; two MPI ranks per core; computed on magnitUDE.

Localized Nonlinearities in 3D								
$p = 4$; $H/h = 30$; exact FETI-DP; computed on magnitUDE								
MPI ranks	Problem size	Nonlinear solver	Local factor.	Coarse factor.		Krylov it.	Execution time	Parallel eff.
				in.	out.			
96	2 650 021	NK	17	-	17	804	464.92s	65%
		NL-1	22	16	6	278	337.06s	89%
		NL-2	23	20	3	146	302.91s	99%
		NL-3	24	0	3	150	300.41s	100%
		NL-4	30	0	4	161	377.07s	80%
768	20 967 241	NK	22	-	22	805	786.32s	38%
		NL-1	27	20	7	580	480.00s	63%
		NL-2	29	25	4	319	441.04s	68%
		NL-3	31	0	4	308	446.72s	67%
		NL-4	36	0	4	299	418.66s	72%
6 144	166 811 281	NK	27	-	27	2 437	1 085.29s	28%
		NL-1	31	24	7	689	587.34s	51%
		NL-2	33	29	4	377	540.86s	56%
		NL-3	36	0	5	396	540.25s	56%
		NL-4	41	0	4	344	490.25s	61%

TABLE 7

Model problem *Localized Nonlinearities in 3D* (see subsection 5.1). New nonlinear FETI-DP algorithms compared to the more traditional Newton–Krylov–FETI-DP; we use the same settings as in Table 6 except for $\Omega = (0, 4) \times (0, 4) \times (0, 3)$; computed on magnitUDE.

Localized Nonlinearities in 3D								
$p = 4$; $H/h = 30$; exact FETI-DP; computed on magnitUDE								
MPI ranks	Problem size	Nonlinear solver	Local factor.	Coarse factor.		Krylov it.	Execution time	Parallel eff.
				in.	out.			
48	1 332 331	NK	13	-	13	430	316.93s	90%
		NL-1	20	13	7	243	300.59s	95%
		NL-2	24	20	4	110	307.18s	93%
		NL-3	23	0	4	129	286.62s	100%
		NL-4	35	0	6	179	425.14s	67%
384	10 512 661	NK	14	-	14	1 002	668.84s	43%
		NL-1	22	14	8	568	543.11s	53%
		NL-2	25	21	4	263	418.15s	69%
		NL-3	23	0	4	372	381.35s	75%
		NL-4	38	0	6	392	565.94s	51%
3 072	83 521 321	NK	17	-	17	1 560	704.02s	41%
		NL-1	22	14	8	687	495.83s	58%
		NL-2	24	20	4	330	410.92s	70%
		NL-3	22	0	4	364	377.20s	76%
		NL-4	40	0	6	515	565.37s	51%
24 576	665 858 641	NK	17	-	17	1 586	1 634.88s	18%
		NL-1	23	15	8	673	988.49s	29%
		NL-2	24	20	4	357	756.77s	38%
		NL-3	23	0	4	371	613.18s	47%
		NL-4	44	0	6	554	967.76s	30%

tional work. This is shown in Figure 13, where we present the average runtimes per Newton step for the inner loops of the NL-2, NL-3, and NL-4 methods compared to the traditional Newton–Krylov method. A Newton step of the Newton–Krylov method has a cost comparable to an outer Newton step of the NL-1, NL-2, NL-3, or NL-4 method.

It is apparent that the removal of the communication related to the operator B (and B^T) leads to a significant reduction of the average time per Newton step. For

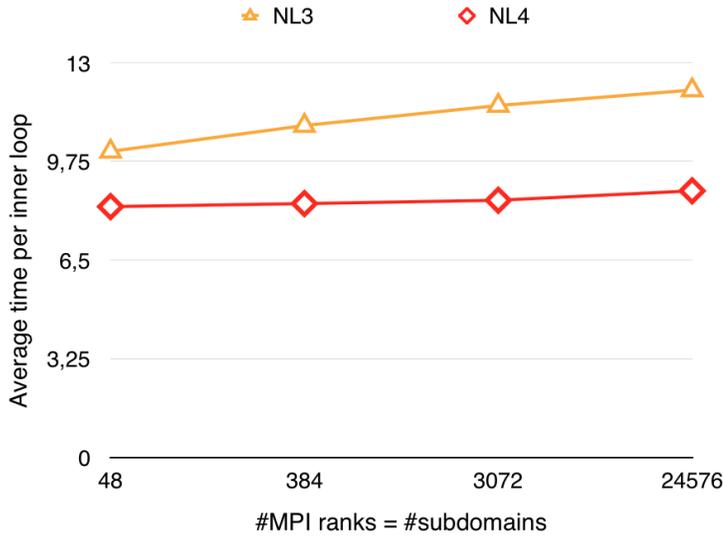


FIG. 11. Model problem Localized Nonlinearities in 3D: Comparison of the weak scalability behavior of the inner loops of NL-3 and NL-4. Here, we present the average runtime per Newton step for computations performed on magnitUDE; see Table 7 for the complete results.

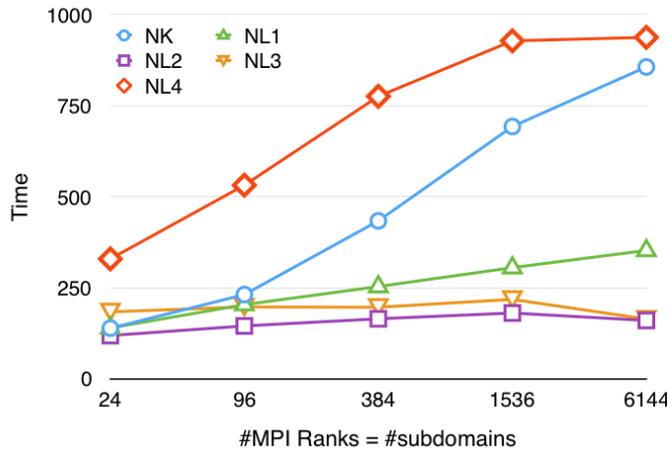


FIG. 12. Model problem Nonlocal Nonlinearities – Channels: Weak scalability of new FETI-DP algorithms and the more traditional Newton–Krylov–FETI-DP algorithm on magnitUDE at Universität Duisburg-Essen; the fastest nonlinear FETI-DP methods (NL-2 and NL-3) are more than five times faster than the traditional NK approach; data from Table 8.

the methods NL-3 and NL-4, a much better parallel scalability is also achieved. For the NL-2 method, the coarse operator (which is factored exactly in this computation) results in a drop of the parallel efficiency for 32 768 cores. This can be avoided by using a multilevel preconditioner for the coarse problem instead of a sparse direct solver; see subsection 2.6. We can conclude that moving computational work from the outer loop of nonlinear methods to the inner loop can reduce the computing times

TABLE 8

Model problem *Nonlocal Nonlinearities – Channels* (see subsection 5.1). New nonlinear FETI-DP algorithms compared to the more traditional Newton–Krylov–FETI-DP; $p = 4$ and a multiplicative weight of $\alpha = 1e5$ inside the channels and $p = 2$ and $\beta = 1$ elsewhere; each subdomain intersected by one channel; width of a channel is $1/2H$; domain $\Omega = (0, 1.5) \times (0, 1)$ decomposed into square subdomains; $H/h = 400$; $\varepsilon_I = 1e-7$; $\varepsilon_O = 1e-8$; two MPI ranks per core; computed on *magnitUDE*; also see Figure 12.

Nonlocal Nonlinearities - Channels								
2D; $\alpha = 1e5$; $p = 4$; $H/h = 400$; exact FETI-DP; computed on <i>magnitUDE</i>								
MPI ranks	Problem size	Nonlinear solver	Local factor.	Coarse factor.		Krylov it.	Execution time	Parallel effic.
				in.	out.			
24	3 844 001	NK	15	-	15	420	138.48s	85%
		NL-1	21	11	10	367	139.23s	85%
		NL-2	30	26	4	119	118.37s	100%
		NL-3	53	0	4	115	183.39s	65%
		NL-4	71	0	12	561	328.94s	36%
96	15 368 001	NK	13	-	13	1 171	230.88s	51%
		NL-1	21	11	10	804	203.36s	58%
		NL-2	30	26	4	265	144.75s	82%
		NL-3	53	0	4	261	196.99s	60%
		NL-4	76	0	13	1 818	531.38s	22%
384	61 456 001	NK	12	-	12	2 553	433.40s	27%
		NL-1	19	10	9	1 193	252.97s	47%
		NL-2	29	25	4	426	164.38s	72%
		NL-3	44	0	4	424	196.16s	60%
		NL-4	62	0	12	3 637	775.52s	15%
1536	245 792 001	NK	11	-	11	4 041	692.55s	17%
		NL-1	19	10	9	1 479	304.93s	39%
		NL-2	28	24	4	534	180.31s	66%
		NL-3	46	0	4	497	217.84s	53%
		NL-4	53	0	12	4 596	927.91s	13%
6144	983 104 001	NK	11	-	11	4 698	856.28s	14%
		NL-1	19	10	9	1 666	352.16s	34%
		NL-2	24	21	3	427	159.84s	74%
		NL-3	31	0	3	385	163.69s	72%
		NL-4	42	0	11	4 445	937.61s	13%

and improve scalability.

6. Conclusion. We presented a framework unifying all known nonlinear FETI-DP and BDDC methods. At the same time, it was shown that these methods can be interpreted as nonlinear right-preconditioned Newton–Krylov methods. We compared the performance of five different variants, including standard exact FETI-DP as well as highly scalable inexact reduced FETI-DP approaches, using our parallel PETSc implementation. For two as well as three dimensional model problems with localized nonlinearities, it was shown that nonlinear methods could be twice as fast as traditional Newton–Krylov–FETI-DP. For certain model problems with nonlocal nonlinearities, the nonlinear methods could even be more than five times as fast. In most cases, the method denoted NL-2 (introduced in [28, 27]) performed best, but for a large number of MPI ranks and subdomains the method denoted NL-3 (introduced recently in [35]) outpaced the other methods. For the methods using nested Newton iterations, we have also presented a strategy to stop the inner Newton iteration early. The resulting algorithms using this approximate nonlinear elimination (NL-ane) proved to be significantly more robust in many cases. Finally, convincing scalability up to 131 072 JUQUEEN BlueGene/Q cores has been presented.

TABLE 9

Model problem *Nonlocal Nonlinearities – Grid* (see subsection 5.1). New nonlinear FETI-DP algorithms controlling the accuracy of the inner Newton iteration compared to the more traditional Newton–Krylov–FETI-DP method and the closely related NL-1 method without computing the initial value; numbers in brackets belong to the runs of the corresponding traditional nonlinear FETI-DP method; div indicates no convergence; $p = 4$ and a weight of $\alpha = 1$ inside the grid and $p = 2$ and $\beta = 1$ elsewhere; see also Figure 8 (left); $\Omega = (0, 1)^2$; decomposed into square subdomains; N is the number of subdomains; $H/h = 16$; $\varepsilon_I = 1e-12$; $\varepsilon_O = 1e-12$; one MPI rank per core; computed on Schwarz.

Nonlocal Nonlinearities – Grid						
2D; $p = 4$; $H/h = 16$; exact FETI-DP; computed on Schwarz						
N	Problem size	Nonlinear solver	Local factor.	Coarse factor.		Krylov it.
				in.	out.	
4	1 089	NK	11	-	11	599
		NL-1 no Init	10	-	10	563
		NL-ane-2	14 (59)	7 (48)	7 (11)	310 (480)
		NL-ane-3	14 (76)	0 (0)	7 (14)	307 (805)
		NL-ane-4	14 (32)	0 (0)	8 (9)	425 (470)
16	4 025	NK	13	-	13	1174
		NL-1 no Init	12	-	12	1148
		NL-ane-2	15 (div)	7 (div)	8 (div)	490 (div)
		NL-ane-3	15 (div)	0 (div)	8 (div)	471 (div)
		NL-ane-4	17 (34)	0 (0)	10 (9)	734 (712)
64	16 641	NK	15	-	15	1891
		NL-1 no Init	14	-	14	1857
		NL-ane-2	15 (div)	9 (div)	6 (div)	576 (div)
		NL-ane-3	15 (div)	0 (div)	9 (div)	803 (div)
		NL-ane-4	21 (44)	0 (0)	13 (12)	1421 (1365)
256	66 049	NK	17	-	17	2692
		NL-1 no Init	16	-	16	2602
		NL-ane-2	18 (div)	11 (div)	7 (div)	840 (div)
		NL-ane-3	18 (div)	0 (div)	11 (div)	1221 (div)
		NL-ane-4	23 (51)	0 (0)	15 (15)	2003 (2092)

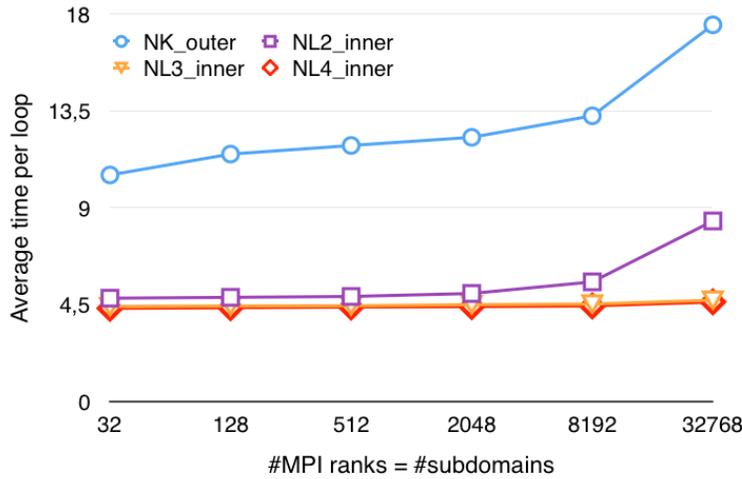


FIG. 13. Model problem *Localized Nonlinearities – Standard Inclusions* (see subsection 5.1): Comparison of the average time spent in the different loops of exact nonlinear FETI-DP versus Newton–Krylov–FETI-DP. The inner loop (NL-2_inner, NL-3_inner, and NL-4_inner) represents the nonlinear elimination step. The timings are for $H/h = 200$; $\varepsilon_I = 1e-6$; $\varepsilon_O = 1e-12$; $\Omega = (0, 2) \times (0, 1)$; computed on JUQUEEN. The computation of the initial value in NL-1 is not shown here, as it has a cost comparable to that of the inner loop in NL-2.

Acknowledgments. The authors gratefully acknowledge the computing time granted by the Center for Computational Sciences and Simulation (CCSS) of the Universität of Duisburg-Essen and provided on the supercomputer magnitUDE (DFG grants INST 20876/209-1 FUGG, INST 20876/243-1 FUGG) at the Zentrum für Informations- und Mediendienste (ZIM).

Computing time on the Taurus HPC Cluster at Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH) in Dresden is gratefully acknowledged.

The authors also gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for providing computing time on the GCS Supercomputer SuperMUC at Leibniz Supercomputing Centre (LRZ, www.lrz.de) and JUQUEEN [51] at Jülich Supercomputing Centre (JSC, www.fz-juelich.de/ias/jsc). GCS is the alliance of the three national supercomputing centres HLRS (Universität Stuttgart), JSC (Forschungszentrum Jülich), and LRZ (Bayerische Akademie der Wissenschaften), funded by the German Federal Ministry of Education and Research (BMBF) and the German State Ministries for Research of Baden-Württemberg (MWK), Bayern (StMWFK), and Nordrhein-Westfalen (MIWF).

REFERENCES

- [1] S. BADIA, A. F. MARTÍN, AND J. PRINCIPE, *Multilevel balancing domain decomposition at extreme scales*, SIAM J. Sci. Comput., 38 (2016), pp. C22–C52, <https://doi.org/10.1137/15M1013511>.
- [2] A. H. BAKER, A. KLAWONN, T. KOLEV, M. LANSER, O. RHEINBACH, AND U. M. YANG, *Scalability of classical algebraic multigrid for elasticity to half a million parallel tasks*, in Software for Exascale Computing—SPPEXA 2013–2015, H.-J. Bungartz, P. Neumann, and E. W. Nagel, eds., Lect. Notes Comput. Sci. Eng. 113, Springer, Berlin, 2016, pp. 113–140, https://doi.org/10.1007/978-3-319-40528-5_6.
- [3] S. BALAY, S. ABHYANKAR, M. F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN, V. ELJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, K. RUPP, B. F. SMITH, S. ZAMPINI, H. ZHANG, AND H. ZHANG, *PETSc Users' Manual*, Tech. report ANL-95/11 - Revision 3.7, Argonne National Laboratory, Lemont, IL, 2016, <http://www.mcs.anl.gov/petsc>.
- [4] S. BALAY, S. ABHYANKAR, M. F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN, V. ELJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, K. RUPP, B. F. SMITH, S. ZAMPINI, H. ZHANG, AND H. ZHANG, *PETSc Web page*, <http://www.mcs.anl.gov/petsc>, 2016.
- [5] F. BORDEU, P.-A. BOUCARD, AND P. GOSSELET, *Balancing domain decomposition with nonlinear relocation: Parallel implementation for laminates*, in Proceedings of the 1st International Conference on Parallel, Distributed and Grid Computing for Engineering, B. H. V. Topping and P. Iványi, eds., Civil-Comp Press, Stirling, UK, 2009, <https://doi.org/10.4203/ccp.90.4>.
- [6] J. H. BRAMBLE AND J. E. PASCIAK, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Math. Comp., 50 (1988), pp. 1–17, <https://doi.org/10.2307/2007912>.
- [7] P. R. BRUNE, M. G. KNEPLEY, B. F. SMITH, AND X. TU, *Composing scalable nonlinear algebraic solvers*, SIAM Rev., 57 (2015), pp. 535–565, <https://doi.org/10.1137/130936725>.
- [8] X.-C. CAI, *Nonlinear overlapping domain decomposition methods*, in Domain Decomposition Methods in Science and Engineering XVIII, Lect. Notes Comput. Sci. Eng. 70, Springer, Berlin, 2009, pp. 217–224, https://doi.org/10.1007/978-3-642-02677-5_23.
- [9] X.-C. CAI AND D. E. KEYES, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183–200, <https://doi.org/10.1137/S106482750037620X>.
- [10] X.-C. CAI, D. E. KEYES, AND L. MARCINKOWSKI, *Non-linear additive Schwarz preconditioners and application in computational fluid dynamics*, Internat. J. Numer. Methods Fluids, 40 (2002), pp. 1463–1470, <https://doi.org/10.1002/flid.404>.
- [11] X.-C. CAI AND X. LI, *Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity*, SIAM J. Sci. Comput., 33 (2011), pp. 746–762, <https://doi.org/10.1137/080736272>.

- [12] J.-M. CROS, *A preconditioner for the Schur complement domain decomposition method*, in Proceedings of the 14th International Conference on Domain Decomposition Methods in Science and Engineering, O. W. I. Herrera, D. Keyes, and R. Yates, eds., National Autonomous University of Mexico (UNAM), Mexico City, Mexico, 2003, pp. 373–380, <http://www.ddm.org/DD14>.
- [13] T. A. DAVIS, *A column pre-ordering strategy for the unsymmetric-pattern multifrontal method*, ACM Trans. Math. Software, 30 (2004), pp. 165–195, <https://doi.org/10.1145/992200.992205>.
- [14] C. R. DOHRMANN, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., 25 (2003), pp. 246–258, <https://doi.org/10.1137/S1064827502412887>.
- [15] V. DOLEAN, M. J. GANDER, W. KHERLJI, F. KWOK, AND R. MASSON, *Nonlinear preconditioning: How to use a nonlinear Schwarz method to precondition Newton's method*, SIAM J. Sci. Comput., 38 (2016), pp. A3357–A3380, <https://doi.org/10.1137/15M102887X>.
- [16] S. C. EISENSTAT AND H. F. WALKER, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32, <https://doi.org/10.1137/0917003>.
- [17] 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*, Internat. J. Numer. Methods Engrg., 50 (2001), pp. 1523–1544.
- [18] C. FARHAT, M. LESOINNE, AND K. PIERSON, *A scalable dual-primal domain decomposition method*, Numer. Linear Algebra Appl., 7 (2000), pp. 687–714, [https://doi.org/10.1002/1099-1506\(200010/12\)7:7/8<687::AID-NLA219>3.0.CO;2-S](https://doi.org/10.1002/1099-1506(200010/12)7:7/8<687::AID-NLA219>3.0.CO;2-S).
- [19] C. FARHAT, J. MANDEL, AND F.-X. ROUX, *Optimal convergence properties of the FETI domain decomposition method*, Comput. Methods Appl. Mech. Engrg., 115 (1994), pp. 367–388, [https://doi.org/10.1016/0045-7825\(94\)90068-X](https://doi.org/10.1016/0045-7825(94)90068-X).
- [20] C. GROSS, *A Unifying Theory for Nonlinear Additively and Multiplicatively Preconditioned Globalization Strategies: Convergence Results and Examples from the Field of Nonlinear Elastostatics and Elastodynamics*, Ph.D. thesis, Rheinische Friedrich-Wilhelms Universität Bonn, Bonn, Germany, 2009.
- [21] C. GROSS AND R. KRAUSE, *A Generalized Recursive Trust-Region Approach—Nonlinear Multiplicatively Preconditioned Trust-Region Methods and Applications*, Tech. report 2010-09, Institute of Computational Science, Universita della Svizzera Italiana, Lugano, Switzerland, 2010.
- [22] C. GROSS AND R. KRAUSE, *On the Globalization of ASPIN Employing Trust-Region Control Strategies—Convergence Analysis and Numerical Examples*, Tech. report 2011-03, Institute of Computational Science, Universita della Svizzera Italiana, Lugano, Switzerland, 2011.
- [23] V. E. HENSON AND U. M. YANG, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177, [https://doi.org/10.1016/S0168-9274\(01\)00115-5](https://doi.org/10.1016/S0168-9274(01)00115-5).
- [24] F.-N. HWANG AND X.-C. CAI, *Improving robustness and parallel scalability of Newton method through nonlinear preconditioning*, in Domain Decomposition Methods in Science and Engineering, Lect. Notes Comput. Sci. Eng. 40, Springer, Berlin, 2005, pp. 201–208, https://doi.org/10.1007/3-540-26825-1_17.
- [25] F.-N. HWANG AND X.-C. CAI, *A class of parallel two-level nonlinear Schwarz preconditioned inexact Newton algorithms*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 1603–1611, <https://doi.org/10.1016/j.cma.2006.03.019>.
- [26] A. KLAWONN, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 172–184, <https://doi.org/10.1137/S1064827596303624>.
- [27] A. KLAWONN, M. LANSER, P. RADTKE, AND O. RHEINBACH, *On an adaptive coarse space and on nonlinear domain decomposition*, in Domain Decomposition Methods in Science and Engineering XXI, J. Erhel, M. J. Gander, L. Halpern, G. Pichot, T. Sassi, and O. Widlund, eds., Lect. Notes Comput. Sci. Eng. 98, Springer, Berlin, 2014, pp. 71–83, https://doi.org/10.1007/978-3-319-05789-7_6.
- [28] A. KLAWONN, M. LANSER, AND O. RHEINBACH, *Nonlinear FETI-DP and BDDC methods*, SIAM J. Sci. Comput., 36 (2014), pp. A737–A765, <https://doi.org/10.1137/130920563>.
- [29] A. KLAWONN, M. LANSER, AND O. RHEINBACH, *Exasteel—computational scale bridging using a FE²TI approach with ex_{nl}/FE²*, in JUQUEEN Extreme Scaling Workshop 2015, D. Brömmel, W. Frings, and B. J. N. Wylie, eds., FZJ-JSC-IB-2015-01, 2015, pp. 15–21, <https://user.fz-juelich.de/record/188191>.

- [30] A. KLAWONN, M. LANSER, AND O. RHEINBACH, *A nonlinear FETI-DP method with an inexact coarse problem*, in Domain Decomposition Methods in Science and Engineering XXII, R. Krause, M. Gander, T. Dickopf, L. Pavarino, and L. Halpern, eds., Lect. Notes Comput. Sci. Eng. 104, Springer, Berlin, 2015, pp. 41–52, https://doi.org/10.1007/978-3-319-18827-0_4.
- [31] A. KLAWONN, M. LANSER, AND O. RHEINBACH, *Toward extremely scalable nonlinear domain decomposition methods for elliptic partial differential equations*, SIAM J. Sci. Comput., 37 (2015), pp. C667–C696, <https://doi.org/10.1137/140997907>.
- [32] A. KLAWONN, M. LANSER, AND O. RHEINBACH, *FE²TI: Computational scale bridging for dual-phase steels*, in Parallel Computing: On the Road to Exascale, Adv. Parallel Comput. 27, IOS Press, Amsterdam, The Netherlands, 2016, pp. 797–806, <https://doi.org/10.3233/978-1-61499-621-7-797>.
- [33] A. KLAWONN, M. LANSER, AND O. RHEINBACH, *A highly scalable implementation of inexact nonlinear FETI-DP without sparse direct solvers*, in Numerical Mathematics and Advanced Applications ENUMATH 2015, Lect. Notes Comput. Sci. Eng. 112, Springer, Berlin, 2016, pp. 255–264, https://doi.org/10.1007/978-3-319-39929-4_25.
- [34] A. KLAWONN, M. LANSER, O. RHEINBACH, AND M. URAN, *Choosing the accuracy of the inner Newton iteration in nonlinear domain decomposition*, submitted to the Proceedings of the 24th International Conference on Domain Decomposition, Lect. Notes Comput. Sci. Eng., Springer, Berlin, 2017.
- [35] A. KLAWONN, M. LANSER, O. RHEINBACH, AND M. URAN, *New nonlinear FETI-DP methods based on a partial nonlinear elimination of variables*, in Domain Decomposition Methods in Science and Engineering XXIII, Lect. Notes Comput. Sci. Eng. 116, Springer, Cham, 2017, pp. 207–215, https://doi.org/10.1007/978-3-319-52389-7_20.
- [36] A. KLAWONN AND O. RHEINBACH, *Inexact FETI-DP methods*, Internat. J. Numer. Methods Engrg., 69 (2007), pp. 284–307, <https://doi.org/10.1002/nme.1758>.
- [37] A. KLAWONN AND O. RHEINBACH, *Robust FETI-DP methods for heterogeneous three dimensional elasticity problems*, Comput. Methods Appl. Mech. Engrg., 196 (2007), pp. 1400–1414, <https://doi.org/10.1016/j.cma.2006.03.023>.
- [38] A. KLAWONN AND O. RHEINBACH, *Highly scalable parallel domain decomposition methods with an application to biomechanics*, ZAMM Z. Angew. Math. Mech., 90 (2010), pp. 5–32, <https://doi.org/10.1002/zamm.200900329>.
- [39] A. KLAWONN AND O. B. WIDLUND, *Dual-primal FETI methods for linear elasticity*, Comm. Pure Appl. Math., 59 (2006), pp. 1523–1572, <https://doi.org/10.1002/cpa.20156>.
- [40] A. KLAWONN, O. B. WIDLUND, AND M. DRYJA, *Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients*, SIAM J. Numer. Anal., 40 (2002), pp. 159–179, <https://doi.org/10.1137/S0036142901388081>.
- [41] D. A. KNOLL AND D. E. KEYES, *Jacobian-free Newton-Krylov methods: A survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397, <https://doi.org/10.1016/j.jcp.2003.08.010>.
- [42] M. LANSER, *Nonlinear FETI-DP and BDDC Methods*, Ph.D. thesis, Universität zu Köln, Köln, Germany, 2015.
- [43] P. J. LANZKRON, D. J. ROSE, AND J. T. WILKES, *An analysis of approximate nonlinear elimination*, SIAM J. Sci. Comput., 17 (1996), pp. 538–559, <https://doi.org/10.1137/S106482759325154X>.
- [44] J. LI AND O. B. WIDLUND, *FETI-DP, BDDC, and block Cholesky methods*, Internat. J. Numer. Methods Engrg., 66 (2006), pp. 250–271, <https://doi.org/10.1002/nme.1553>.
- [45] L. LIU AND D. E. KEYES, *Field-split preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 37 (2015), pp. A1388–A1409, <https://doi.org/10.1137/140970379>.
- [46] J. MANDEL AND C. R. DOHRMANN, *Convergence of a balancing domain decomposition by constraints and energy minimization*, Numer. Linear Algebra Appl., 10 (2003), pp. 639–659, <https://doi.org/10.1002/nla.341>.
- [47] J. MANDEL, C. R. DOHRMANN, AND R. TEZAUER, *An algebraic theory for primal and dual substructuring methods by constraints*, Appl. Numer. Math., 54 (2005), pp. 167–193, <https://doi.org/10.1016/j.apnum.2004.09.022>.
- [48] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, New York, 2000.
- [49] J. M. ORTEGA AND W. C. RHEINBOLDT, *Iterative Solution of Nonlinear Equations in Several Variables*, Classics Appl. Math. 30, SIAM, Philadelphia, 2000, <https://doi.org/10.1137/1.9780898719468>. Reprint of the 1970 original.
- [50] J. PEBREL, C. REY, AND P. GOSSELET, *A nonlinear dual-domain decomposition method: Application to structural problems with damage*, Internat. J. Multiscale Comput. Engrg., 6 (2008), pp. 251–262, <https://doi.org/10.1615/IntJMultCompEng.v6.i3.50>.

- [51] M. STEPHAN AND J. DOCTER, *JUQUEEN: IBM Blue Gene/Q[®] Supercomputer System at the Jülich Supercomputing Centre*, J. Large-Scale Res. Facilities, 1 (2015), A1, <https://doi.org/10.17815/jlsrf-1-18>.
- [52] A. TOSELLI AND O. WIDLUND, *Domain Decomposition Methods—Algorithms and Theory*, Springer Ser. Comput. Math. 34, Springer-Verlag, Berlin, 2005.
- [53] M. ULBRICH AND S. ULBRICH, *Nichtlineare Optimierung*, Birkhäuser, Basel, 2012, <https://doi.org/10.1007/978-3-0346-0654-7>.