Composing Two Different Nonlinear FETI–DP Methods

Stephan Köhler and Oliver Rheinbach

1 Nonlinear FETI–DP

Nonlinear FETI–DP methods [3] are nonlinear generalizations of linear FETI–DP domain decomposition methods [10]. Using a divide-and-conquer approach, the unconstrained minimization of some objective J is transformed into a constrained optimization problem over many subdomains,

$$\min_{\tilde{u}} \widetilde{J}(\tilde{u}) \quad \text{subject to (s.t.)} \quad B\tilde{u} = 0, \tag{1}$$

where the constraint $B\tilde{u} = 0$ enforces continuity across subdomain boundaries; here, $\tilde{u} := [u_{BB}^{(1)}, \dots, u_{BB}^{(N)}, \tilde{u}_\Pi]^T$, where the subscript B refers to the union of the inner and dual variables, $\widetilde{J}(\tilde{u}) := \sum_{i=1}^N J^{(i)}(u_{BB}^{(i)}, R_\Pi^{(i)} \tilde{u}_\Pi)$, $J^{(i)}$ is the local objective of the i-th subdomain, $R_\Pi^{(i)}$ is the assembly operator of the primal variables as in linear FETI–DP methods [10]. The Lagrange function for (1) is $\mathcal{L}(\tilde{u}, \lambda) = \widetilde{J}(\tilde{u}) + \lambda^T B\tilde{u}$. The saddle point problem of the first-order necessary optimality condition

$$\begin{aligned} \nabla_{\tilde{u}} \mathcal{L}(\tilde{u}, \lambda) &= \nabla \widetilde{J}(\tilde{u}) + B^T \lambda = \tilde{f}, \\ \nabla_{\lambda} \mathcal{L}(\tilde{u}, \lambda) &= \qquad\qquad\quad B\tilde{u} = 0, \end{aligned} \tag{2}$$

corresponds directly to the linear FETI–DP saddle point problem [10]. The nonlinear operator $\nabla \widetilde{J}(\tilde{u}) := R_\Pi^T \nabla J(R_\Pi \tilde{u})$ is obtained from finite element subassembly of the blocks $\nabla J^{(i)}(u_{BB}^{(i)}, R_\Pi^{(i)} \tilde{u}_\Pi)$ in the primal variables using the operator R_Π^T as in linear FETI–DP methods [10]. This coupling provides a nonlinear coarse problem for the method. Thus, $\nabla \widetilde{J}$ represents a nonlinear coarse approximation of the original problem.

Stephan Köhler, Oliver Rheinbach
Institut für Numerische Mathematik und Optimierung, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, 09596 Freiberg,
e-mail: oliver.rheinbach@math.tu-freiberg.de, stephan.koehler@math.tu-freiberg.de

Next, we perform the nonlinear elimination: we split the first row in (2) according to disjoint index sets E, L (eliminate or linearize) and solve in a first step

$$\nabla_{\tilde{u}_E}\mathcal{L}(\tilde{u}_E, \tilde{u}_L, \lambda) = \nabla_{\tilde{u}_E}\widetilde{J}(\tilde{u}_E, \tilde{u}_L) + B_E^T\lambda = 0, \tag{3}$$

for $\tilde{u}_E$, given $\tilde{u}_L$ and λ. Then, we can insert $\tilde{u}_E$ into the remaining equations and solve by linearization in $\tilde{u}_L$ and λ, and using the implicit function theorem. In [3], four different elimination sets are considered: Nonlinear FETI–DP-1 (*NL*-1), where $E = \emptyset$, nonlinear FETI–DP-2 (*NL*-2), where E contains all variables and $L = \emptyset$, nonlinear FETI–DP-3 (*NL*-3), where E contains the inner and the dual variables, and Nonlinear FETI–DP-4 (*NL*-4), where E contains only the inner variables. Here, we focus on the two elimination sets of *NL*-2 and *NL*-4.

2 Composing two different nonlinear FETI–DP methods

We combine the two different nonlinear FETI–DP methods *NL*-2, where all variables are eliminated, and *NL*-4 where only the inner variables are eliminated. The idea is based on [9], where a similar approach is successfully applied for nonlinear FETI–1. In a first step, for given multipliers $\lambda^{(k)}$, the implicit function $g_1(\lambda^{(k)})$, such that

$$\nabla_{\tilde{u}}\mathcal{L}\big|_{(g_1(\lambda^{(k)}),\lambda^{(k)})} = 0, \tag{4}$$

where we denote the evaluation of $\nabla_{\tilde{u}}\mathcal{L}$ at the point $(g_1(\lambda^{(k)}), \lambda^{(k)})$ by $\nabla_{\tilde{u}}\mathcal{L}\big|_{(g_1(\lambda^{(k)}),\lambda^{(k)})}$, is computed. The function g_1 corresponds to *NL*-2. Afterwards, we compute a weighted average over the interface by

$$g_2(\lambda^{(k)}) := (I - B^{*T} B) g_1(\lambda^{(k)}), \tag{5}$$

where B^* is a pseudo-inverse for B such that $B\, B^{*T} B = B$ and $B^* B^T B^* = B^*$. A cost-saving variant for B^* is $B^* = B_{D,\Gamma}$, where $B_{D,\Gamma}$ corresponds to the Dirichlet preconditioner of the standard FETI–DP method; see, e.g., [8]. Due to (5), it follows immediately that $Bg_2(\lambda^{(k)}) = 0$. However, there is an unnatural tension between the interface variables of g_2 and the variables adjacent to them. To resolve this tension, in a third step, we compute the implicit function corresponding to *NL*-4,

$$g_3(\lambda^{(k)}) := \left(h(g_{2,\Delta}(\lambda^{(k)}), g_{2,\Pi}(\lambda^{(k)}), \lambda^{(k)})^T, \, g_{2,\Delta}(\lambda^{(k)})^T, \, g_{2,\Pi}(\lambda^{(k)})^T\right)^T, \tag{6}$$

where $g_{2,\Delta}$, $g_{2,\Pi}$ are the dual, primal variables of g_2, respectively, and the implicit function $h(g_{2,\Delta}(\lambda^{(k)}), g_{2,\Pi}(\lambda^{(k)}), \lambda^{(k)})$ solves

$$\nabla_I \mathcal{L}\big|_{\left(h\left(g_{2,\Delta}(\lambda^{(k)}),g_{2,\Pi}(\lambda^{(k)}),\lambda^{(k)}\right),\, g_{2,\Delta}(\lambda^{(k)}),\, g_{2,\Pi}(\lambda^{(k)}),\, \lambda^{(k)}\right)} = 0, \tag{7}$$

where we denote the gradient with respect to the inner variables of $\tilde{u}$ by ∇_I.

We obtain the new multipliers by $\lambda^{(k+1)} = -B^T B^* \nabla \widetilde{J}\big|_{g_3(\lambda^{(k)})}$, for details, see [9]. If $(\tilde{u}^*, \lambda^*)$ is a KKT point of (1) and if $\nabla_{\tilde{u}\tilde{u}} \mathcal{L}\big|_{(\tilde{u}^*, \lambda^*)}$ is invertible, it follows from (4) by the implicit function theorem that $g_1(\lambda^*) = \tilde{u}^*$. Furthermore, it follows that $B\tilde{u}^* = 0$ and therefore, from (4) and (5), we have $g_1(\lambda^*) = g_2(\lambda^*)$. From (4), it follows that $g_1(\lambda^*) = g_3(\lambda^*)$. By the first part of the first-order necessary optimality condition (2), we have $\nabla \widetilde{J}(\tilde{u}^*) = -B^T \lambda^*$. Therefore, the nonlinear root-finding problem is

$$r(\lambda) := -B^T B^* \nabla \widetilde{J}\big|_{g_3(\lambda^{(k)})} - B^T \lambda = -B^T B^* \nabla_{\tilde{u}} \mathcal{L}\big|_{g_3(\lambda^{(k)})}. \tag{8}$$

Since we assume that B^T has full rank, which can always guaranteed by the use of nonredundant multipliers, we can rewrite (8) as $r(\lambda) := B^* \nabla_{\tilde{u}} \mathcal{L}\big|_{g_3(\lambda^{(k)})}$. We apply Newton's method to $r(\lambda)$. By similar arguments as outlined in [9], we have

$$Dr(\lambda) \approx \left(B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)} B^{*T}_{\widetilde{\Gamma}} \, B_{\widetilde{\Gamma}} \, S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)}^{-1} \, B^T_{\widetilde{\Gamma}} \right), \tag{9}$$

where

$$S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)} := \\ \left. \begin{pmatrix} \nabla^2_{\Delta\Delta}\mathcal{L} - \nabla^2_{\Delta I}\mathcal{L}\, \nabla^2_{II}\mathcal{L}^{-1}\, \nabla^2_{I\Delta}\mathcal{L} & \nabla^2_{\Delta\Pi}\mathcal{L} - \nabla^2_{\Delta I}\mathcal{L}\, \nabla^2_{II}\mathcal{L}^{-1}\, \nabla^2_{I\Pi}\mathcal{L} \\ \nabla^2_{\Pi\Delta}\mathcal{L} - \nabla^2_{\Pi I}\mathcal{L}\, \nabla^2_{II}\mathcal{L}^{-1}\, \nabla^2_{I\Delta}\mathcal{L} & \nabla^2_{\Pi\Pi}\mathcal{L} - \nabla^2_{\Pi I}\mathcal{L}\, \nabla^2_{II}\mathcal{L}^{-1}\, \nabla^2_{I\Pi}\mathcal{L} \end{pmatrix} \right|_{(g_3(\lambda),\lambda)}$$

and $B^*_{\widetilde{\Gamma}}$, $B_{\widetilde{\Gamma}}$ correspond to the interface part of B^*, B, respectively. Let us remark that the approximation (9) uses $g_1(\lambda) \approx g_3(\lambda)$.

The operator in (9) consists of two parts: the FETI–DP system matrix $B_{\widetilde{\Gamma}} \, S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)}^{-1} \, B^T_{\widetilde{\Gamma}}$ and the Dirichlet preconditioner $B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)} B^{*T}_{\widetilde{\Gamma}}$; for the linear case, see, e.g., [8, 10] and for the nonlinear case, see, e.g.,[3, 7]. The Newton equation for $r(\lambda)$ is given by

$$\left(B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}} B^{*T}_{\widetilde{\Gamma}} \, B_{\widetilde{\Gamma}} \, S_{\widetilde{\Gamma}\widetilde{\Gamma}}^{-1} \, B^T_{\widetilde{\Gamma}} \right)\Big|_{(g_3(\lambda),\lambda)} \delta\hat{\lambda} = -B^* \nabla_{\tilde{u}} \mathcal{L}\big|_{(g_3(\lambda),\lambda)}. \tag{10}$$

The system matrix in (10) corresponds to the system matrix of a standard preconditioned FETI-DP system, however, the preconditioner $B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}} B^{*T}_{\widetilde{\Gamma}}$ is not applied to the right hand side, which is an important difference.

For completeness, we show the preconditioned Newton equation for *NL*-2:

$$\left(B^{*}_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}} B^{*T}_{\widetilde{\Gamma}}\, B_{\widetilde{\Gamma}}\, {S_{\widetilde{\Gamma}\widetilde{\Gamma}}}^{-1}\, B^{T}_{\widetilde{\Gamma}}\right)\Big|_{(g_1(\lambda),\lambda)} \delta\lambda = -B^{*}_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}}\Big|_{(g_1(\lambda),\lambda)} B^{*T}_{\widetilde{\Gamma}}\, B g_1(\lambda).$$

Note, the difference between the two equations is the evaluation point of the operator and the right hand side.

3 Globalization of nonlinear FETI–DP

For the globalization of the method outlined in Section 2, we use the exact differentiable penalty function

$$P(\tilde{u},\lambda;M,\mu) = \mathcal{L}(\tilde{u},\lambda) + \frac{\mu}{2}\|c(\tilde{u})\|^2 + \frac{1}{2}\|M\nabla_{\tilde{u}}\mathcal{L}(\tilde{u},\lambda)\|^2 \tag{11}$$

introduced in [2]. For a detailed analysis of P, we refer to [1]. For nonlinear FETI–DP, we have $c(\tilde{u}) = B\tilde{u}$ and $M = \eta\, B$, where B is the FETI–DP jump operator. First results for globalization of nonlinear FETI–DP by P were presented in [6] and for a detailed analysis we refer to [5]. The methods presented in [6] make explicit use of the nonlinear elimination. Indeed, the nonlinear elimination needs to be computed in every step of the backtracking. Such an approach for the function g_3 from Section 2 is computationally expensive also when computing the exact Jacobian Dg_3. Let us keep in mind that (9) uses the approximation $g_3(\lambda) \approx g_1(\lambda)$.

Simplified backtracking Hence, we need to modify the globalization approach. The main idea is that for given point $(\tilde{u}^{(k)},\lambda^{(k)})$ we compute a new trial point $(g_3(\hat{\lambda}^{(k)}),\hat{\lambda}^{(k)})$, where $\hat{\lambda}^{(k)} = \lambda^{(k)} + \delta\hat{\lambda}^{(k)}$ and $\delta\hat{\lambda}^{(k)}$ is the solution of (10) at $(g_3(\lambda^{(k)}),\lambda^{(k)})$. Afterwards, we compute our search direction by

$$d_1^{(k)} = \left((g_3(\hat{\lambda}^{k}) - \tilde{u}^{(k)})^T,\, {\delta\hat{\lambda}^{(k)}}^T\right)^T. \tag{12}$$

Similarly as for a Newton-direction, it is unclear if $d_1^{(k)}$ is a descent direction. Therefore, we must ensure that a generalized angle condition holds if we use $d_1^{(k)}$. If $d_1^{(k)}$ does not fulfill a generalized angle condition, i.e.,

$$\nabla\, {P^{(k)}}^T\, d_1^k \geq -\min\{\eta_1, \eta_2\, \|\nabla P^{(k)}\|_\infty^p\}\, \|d_1^{(k)}\|_2\, \|\nabla P^{(k)}\|_\infty, \tag{13}$$

where $\nabla P^{(k)} := \nabla P\big|_{(\tilde{u}^{(k)},\lambda^{(k)};M,\mu_k)}$ or if

$$\|d_1^{(k)}\|_2 < \eta_3\left(-\nabla\, {P^{(k)}}^T\, d_1^{(k)}\right)/\|d_1^{(k)}\|_2, \tag{14}$$

we compute a new direction $d_2^{(k)}$ by the solution of the standard Lagrange-Newton equation at the point $(\tilde{u}^{(k)},\lambda^{(k)})$. Let us remark that the solution of the Lagrange-Newton equation correspond to a Newton-like search direction for P; see, e.g., [1].

```
Init: $(\tilde{u}^{(0)}, \lambda^{(0)})$, $\eta_1, \rho \in (0,1)$, $\varepsilon_{\text{update}} > 1$, $\varepsilon_{\text{tol}}, \mu_0, \eta_2, \eta_3, p > 0$.
for $k = 0, 1, \ldots$ until convergence do
  1. If $\|\nabla \mathcal{L}^{(k)}\|_\infty \leq \varepsilon_{\text{tol}}$, STOP.
  2. (a) Compute $g_3(\lambda^{(k)})$    // Includes computation of NL-2 and NL-4.
     (b) Solve $B^*_{\widetilde{\Gamma}} S^{(k)}_{\widetilde{\Gamma}\widetilde{\Gamma}} B^{*T}_{\widetilde{\Gamma}} B_{\widetilde{\Gamma}} {S^{(k)}_{\widetilde{\Gamma}\widetilde{\Gamma}}}^{-1} B^T_{\widetilde{\Gamma}} \delta\hat{\lambda}^{(k)} = -B^* \nabla_{\tilde{u}} \mathcal{L}^{(k)}$.
     (c) Compute $g_3(\lambda^{(k)} + \delta\hat{\lambda}^{(k)})$    // Includes computation of NL-2 and NL-4.
     (d) Set $d^{(k)} = \begin{pmatrix} g_3(\lambda^{(k)} + \delta\hat{\lambda}^{(k)}) - \tilde{u}^{(k)} \\ \delta\hat{\lambda}^{(k)} \end{pmatrix}$.
     if (13) or (14) then
         Set    $d^{(k)} = -\nabla^2 \mathcal{L}^{-1}\big|_{(\tilde{u}^{(k)}, \lambda^{(k)})} \nabla \mathcal{L}\big|_{(\tilde{u}^{(k)}, \lambda^{(k)})}$
         if (13) then
             Set    $d^{(k)} = -\nabla^2 P\big|_{(\tilde{u}^{(k)}, \lambda^{(k)})}$
         end
     end
  3. Compute the step length $\alpha_k$ based on the Armijo rule.
  4. Set $\tilde{u}^{(k+1)} = \tilde{u}^{(k)} + \alpha_k d^{(k)}_{\tilde{u}}$ and $\lambda^{(k+1)} = \lambda^{(k)} + \alpha_k d^{(k)}_{\lambda}$,
     where $d^{(k)} = (d^{(k)T}_{\tilde{u}}, d^{(k)T}_{\lambda})^T$.
  5. if $\| B\tilde{u}^{(k+1)} \| \geq \rho \| B\tilde{u}^{(k)} \|$ then
         Set $\mu_{k+1} = \varepsilon_{\text{update}} \mu_k$.
     else
         Set $\mu_{k+1} = \mu_k$.
     end
end
```

Fig. 1: Minimization algorithm for P.

If $d_2^{(k)}$ does also not fulfill the generalized angle condition (13), we use $-\nabla P^{(k)}$ as the search direction. Afterwards, we compute the step length by the Armijo rule. In contrast to the algorithm outlined in [6], we do not compute any nonlinear elimination in the backtracking of the Armijo rule. This is important to reduce the runtime, since the computation of $g_3(\lambda^{(k)})$ takes some effort. We refer to this as *simplified backtracking*.

Minimization algorithm We outline our minimization algorithm in Fig. 1. Let us explain some details: Our framework is based on a general line search algorithm and a globalized Newton line search algorithm; for details, see, e.g., [11]. We use the simplified backtracking to save some runtime. Furthermore, we do not rely on the exact computation of $g_3(\lambda^{(k)})$, since we are only interested in a descent for P. Therefore, we can abort the computation of the $g_1(\lambda^{(k)})$, which corresponds to the nonlinear elimination of *NL*-2, after a few iterations. This holds also for the nonlinear elimination corresponding to the *NL*-4 step. By such an inexact nonlinear elimination, we try to avoid over solving of the nonlinear elimination problems and the idea is based on the approach in [4]. The inexact nonlinear elimination is an important difference to Newton's method applied to $r(\lambda)$, which needs, formally, the exact computation of $g_3(\lambda^{(k)})$. The disadvantage in Fig. 1 is that if Fig. 1 2. (b) does not provide a descent direction, we have to compute and factorize $\nabla^2\mathcal{L}$ again at the old point $(\tilde{u}^{(k)}, \lambda^{(k)})$, which takes some additional runtime.

Let us remark that the globalization strategy in Fig. 1 combined with an inexact computation of g_3 is based on the ideas in [5].

4 Numerical results

We consider red a two-dimensional beam bending benchmark problem with a Neo-Hookean constitutive law using no or almost incompressible inclusions embedded in each subdomain. The strain energy density function for the compressible matrix material part is given by $J(x) = \frac{\mu}{2}(\mathrm{tr}(F(x)^T F(x)) - 2) - \mu \log(\psi(x)) + \frac{\lambda}{2}(\log(\psi(x)))^2$, where $\psi(x) = \det(F(x))$, $F(x) = \nabla\varphi(x)$, $\varphi(x) = x + u(x)$, $u(x)$ denotes the displacement and μ and λ are the Lamé constants. The nearly incompressible part is given by $J(x) = \frac{\mu}{2}(\mathrm{tr}(\frac{1}{\psi(x)} F(x)^T F(x)) - 2) + \frac{\kappa}{2}(\psi(x) - 1)^2$, where $\kappa = \frac{\lambda(1+\mu)}{3\mu}$; see, e.g. [12, 5] and references therein. As material parameters, we use $E = 210$ and $\nu = 0.3$ for the matrix material and $E = 210$ and $\nu = 0.499$ for the (mildly) almost incompressible inclusions. For the discretization, we use $P2$ elements, which are not stable for the incompressible case.

For the computation of $g_1(\lambda^{(k)})$ in Fig. 1 (a), we solve the minimization problem $\min_{\tilde{u}} \mathcal{L}(\tilde{u}, \lambda^{(k)})$. We solve this problem inexactly, in the sense that we abort the computation if $\frac{|\mathcal{L}(\tilde{u}_{\ell+1}^{(k)}, \lambda^{(k)}) - \mathcal{L}(\tilde{u}_{\ell}^{(k)}, \lambda^{(k)})|}{|\mathcal{L}(\tilde{u}_{\ell}^{(k)}, \lambda^{(k)})|} < \gamma_1$ or if $(1 - \gamma_2) < \frac{\|\nabla_{\tilde{u}} \mathcal{L}(\tilde{u}_{\ell+1}^{(k)}, \lambda^{(k)})\|_\infty}{\|\nabla_{\tilde{u}} \mathcal{L}(\tilde{u}_{\ell}^{(k)}, \lambda^{(k)})\|_\infty}$, where $\tilde{u}_{\ell+1}^{(k)}$ is the current iterate in the computation of $g_1(\lambda^{(k)})$ and $\tilde{u}_{\ell}^{(k)}$ is the previous one. In a similar way, we compute also the NL-4 part of $g_3(\lambda^{(k)})$ inexactly. Let us remark that we use a globalized Newton method with a computation of the Newton step using a direct sparse solver for the Newton equation. This can be afforded since this is an operation local to the subdomains and for g_1 this involves also the (small) coarse space.

As Krylov methods in Fig. 1, we use GMRES. In Table 1, we show the number of (outer) iterations for the $2D$ Neo-Hookean beam bending benchmark problem with a homogeneous material model, see upper part of Table 1, and (mildly) almost incompressible inclusions, see lower part of Table 1. We report the iterations for the standard nonlinear FETI–DP methods NL-1, NL-2, and NL-2 with simplified backtracking, which includes also the inexact nonlinear elimination and the new approach outlined in section 2, NL-2/4, with simplified backtracking and inexact nonlinear elimination. In brackets, we show the cumulative iterations for the nonlinear elimination corresponding to the NL-2 elimination set, we refer to this as inner iterations. This does not include the iterations for the nonlinear elimination corresponding to the NL-4 part in NL-2/4.

In the upper part of Table 1 shows that there is a small increase in the number of outer iterations for NL-2 simpl. compared to the standard NL-2 method; the number of inner iterations, however, decreases significantly. For NL-2/4 simpl., the increase of the number of outer iterations is more significant, but the number of inner iterations stays the same or decreases slightly, again, compared with the NL-2 method. Let us remark that for 4 000 subdomains and NL-2, we need 6 outer iterations instead of the previous 2, since we need to make 4 gradient steps, which are not as effective as Newton steps.

Table 1: Nonlinear FETI–DP-1, 2 (NL-1, 2), nonlinear FETI–DP-2 with simplified backtracking (NL-2 simpl.), and nonlinear FETI–DP-2/4 with simplified backtracking (NL-2/4 simpl.); $H/h \approx 21$; beam bending problem in $2D$; coarse space: vertices, edge, and rotational averages; globalization based on P and using Fig. 1 for NL-2 linear. and NL-2/4 linear. for NL-1, 2 see [5, 6]; start penalty parameter $\mu_0 = 100$; exact diff penalty method; number of iteration is shown, in brackets the cumulative number of nonlinear elimination steps for the NL-2 part; stopping criterion: $\|\nabla \mathcal{L}^{(k)}\|_\infty < 10^{-6}$.

using globalization: NL-1,2, see [6, 5]; NL-2 linear., NL-2-4 linear. see Fig. 1					
homogeneous Neo-Hooke					
		body force $f = (0, -1.0)^T$			
		Standard Methods		New Methods	
#d.o.f.	#Sub.	NL-1	NL-2	NL-2 simpl.	NL-2/4 simpl.
963 202	250	16	2 [16]	3 [11]	5 [16]
3 844 392	1 000	15	1 [15]	3 [11]	5 [16]
15 360 772	4 000	15	1 [15]	3 [11]	5 [15]
		body force $f = (0, -2.0)^T$			
		Standard Methods		New Methods	
#d.o.f.	#Sub.	NL-1	NL-2	NL-2 simpl.	NL-2/4 simpl.
963 202	250	17	2 [19]	4 [14]	6 [17]
3 844 392	1 000	17	2 [19]	4 [14]	6 [17]
15 360 772	4 000	17	6 [28]	3 [13]	6 [18]
incomp. inclusions ($\nu = 0.499$)					
		body force $f = (0, -1.0)^T$			
		Standard Methods		New Methods	
#d.o.f.	#Sub.	NL-1	NL-2	NL-2 simpl.	NL-2/4 simpl.
963 202	250	36	2 [36]	9 [26]	13 [33]
3 844 392	1 000	36	2 [38]	9 [25]	13 [34]
15 360 772	4 000	35	2 [38]	8 [24]	13 [35]
		body force $f = (0, -2.0)^T$			
		Standard Methods		New Methods	
#d.o.f.	#Sub.	NL-1	NL-2	NL-2 simpl.	NL-2/4 simpl.
963 202	250	43	2 [46]	12 [30]	16 [41]
3 844 392	1 000	44	2 [46]	11 [32]	15 [42]
15 360 772	4 000	44	2 [48]	11 [30]	16 [41]

In the lower part of Table 1, we observe a significant increase of the outer iterations for NL-2 simpl. and NL-2/4 simpl. compared to NL-2, but the number of inner iterations descreases significantly for NL-2 simpl. and there is a small improvement for NL-2/4 simpl.

In our experiments, NL-2 simpl. seems to be a give better results than NL-2/4, which was not expected; we suspect that this is due to our large coarse space which includes edge averages and rotations. Hence, the jump at the interface is not large and therefore the correction part on the interface, which includes NL-4, does not lead to a significant improvement. We guess that this will change if the jump of the interface is larger. Note, however, that using a smaller coarse space resulted in very ill-conditioned tangent systems, i.e., the number Krylov iterations was high.

References

1. Bertsekas, D. P. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich, Publishers] New York-London (1982).
2. Di Pillo, G. and Grippo, L. A new class of augmented Lagrangians in nonlinear programming. *SIAM J. Control Optim.* **17**(5), 618–628 (1979).
3. Klawonn, A., Lanser, M., Rheinbach, O., and Uran, M. Nonlinear FETI–DP and BDDC methods: a unified framework and parallel results. *SIAM J. Sci. Comput.* **39**(6), C417–C451 (2017).
4. Klawonn, A., Lanser, M., Rheinbach, O., and Uran, M. On the accuracy of the inner newton iteration in nonlinear domain decomposition. In: *International Conference on Domain Decomposition Methods*, 435–443. Springer (2017).
5. Köhler, S. *Globalization of Nonlinear FETI–DP Methods*. Ph.D. thesis, Technische Universität Bergakademie Freiberg (submitted).
6. Köhler, S. and Rheinbach, O. Globalization of Nonlinear FETI–DP Methods. In: *Domain Decomposition Methods in Science and Engineering XXVI*, *Lect. Notes Comput. Sci. Eng.*, vol. 145, 327–334. Springer Cham (2022).
7. Lanser, M. H. *Nonlinear FETI-DP and BDDC Methods*. Ph.D. thesis, Universität zu Köln (2015).
8. Li, J. and Widlund, O. B. FETI-DP, BDDC, and block cholesky methods. *Internat. J. Numer. Methods Engrg.* **66**(2), 250–271 (2006).
9. Negrello, C., Gosselet, P., and Rey, C. Nonlinearly preconditioned FETI solver for substructured formulations of nonlinear problems. *Mathematics* **9**(24), 3165 (2021).
10. Toselli, A. and Widlund, O. *Domain decomposition methods—algorithms and theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer, Berlin (2005).
11. Ulbrich, M. and Ulbrich, S. *Nichtlineare Optimierung*. Birkhäuser Basel (2012).
12. Uran, M. *High-Performance Computing Two-Scale Finite Element Simulations of a Contact Problem Using Computational Homogenization*. Ph.D. thesis, Universität zu Köln (2020).