

# Combining Nonlinear FETI-DP Methods and Quasi-Newton Methods using an SQP Approach

Stephan Köhler<sup>[0000-0003-1015-8736]</sup>  
and Oliver Rheinbach<sup>[0000-0002-9310-8533]</sup>

## 1 Nonlinear FETI-DP

Nonlinear FETI-DP methods [?] are nonlinear generalizations of standard linear dual-primal Finite Element Tearing and Interconnecting domain decomposition methods [?] based on a divide-and-conquer approach. We consider an unconstrained minimization problem for some objective  $J$  and decompose it following the FETI-DP strategy. This unconstrained problem is transformed into a constrained one by tearing the computational domain at the interface into subdomains, using subassembly for the primal degrees of freedom (dofs) of the subdomain boundaries, and equality constraints for the remaining dofs of the subdomain boundaries, the so called dual variables. The resulting problem can be written as

$$\min_{\tilde{u}} \tilde{J}(\tilde{u}) \quad \text{subject to (s.t.)} \quad B\tilde{u} = 0, \quad (1.1)$$

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

---

Stephan Köhler  
Technische Universität Bergakademie Freiberg, Akademiestr. 6, 09596 Freiberg, e-mail: [stephan.koehler@math.tu-freiberg.de](mailto:stephan.koehler@math.tu-freiberg.de)  
Oliver Rheinbach  
Technische Universität Bergakademie Freiberg, Akademiestr. 6, 09596 Freiberg, e-mail: [oliver.rheinbach@math.tu-freiberg.de](mailto:oliver.rheinbach@math.tu-freiberg.de)

$$\begin{aligned}\nabla_{\tilde{u}}\mathcal{L}(\tilde{u}, \lambda) &= \nabla\tilde{J}(\tilde{u}) + B^T\lambda = \tilde{f}, \\ \nabla_{\lambda}\mathcal{L}(\tilde{u}, \lambda) &= B\tilde{u} = 0.\end{aligned}\tag{1.2}$$

The coarse problem of the method is nonlinear and is obtained from finite element subassembly  $\nabla\tilde{J}(\tilde{u}) := R_{\Pi}^T\nabla J(R_{\Pi}\tilde{u})$ , as in linear FETI-DP methods [?].

## 2 A Sequential Quadratic Programming Approach

We consider the sequential quadratic programming (SQP) approach for the constrained optimization problem (??), see, e.g., [?, ?, ?]. The original constrained problem is replaced by a sequence of quadratic problems of the form

$$\begin{aligned}\min_d \quad & \nabla_{\tilde{u}}\mathcal{L}(\tilde{u}^{(k)}, \lambda^{(k)})^T d + \frac{1}{2}d^T \nabla_{\tilde{u}\tilde{u}}^2\mathcal{L}(\tilde{u}^{(k)}, \lambda^{(k)})d \\ \text{s.t.} \quad & B(\tilde{u}^{(k)} + d) = 0,\end{aligned}\tag{2.1}$$

where  $(\tilde{u}^{(k)}, \lambda^{(k)})$  is the current iterate. The next iterate is obtained by

$$\tilde{u}^{(k+1)} = \tilde{u}^{(k)} + \alpha_k \delta\tilde{u}^{(k)}, \quad \lambda^{(k+1)} = \delta\lambda^{(k)}\tag{2.2}$$

where  $(\delta\tilde{u}^{(k)}, \delta\lambda^{(k)})$  is the Karush-Kuhn-Tucker (KKT) point of the SQP subproblem (??) and  $\alpha_k$  is a suitable step length. To determine the step length  $\alpha_k$ , we use the exact nondifferentiable penalty function

$$P_1(\tilde{u}; \mu) = \tilde{J}(\tilde{u}) + \mu\|B\tilde{u}\|_1,\tag{2.3}$$

where  $\mu$  is the penalty parameter. It can be shown that for  $P_1$  the one-sided directional derivative  $DP_1(\tilde{u}; d, \mu)$  exists for all points  $\tilde{u}$  and in all directions  $d$ . Hence, we can use backtracking and the Armijo condition [?]

$$P_1(\tilde{u}^{(k)} + \alpha_k \delta\tilde{u}^{(k)}; \mu) - P_1(\tilde{u}^{(k)}; \mu) \leq \alpha_k DP_1(\tilde{u}; d, \mu)\tag{2.4}$$

to compute the step length  $\alpha_k$ , see, e.g., [?]. Notice, for the directional derivative  $DP_1$  it holds that

$$\begin{aligned}DP_1(\tilde{u}; \delta\tilde{u}^{(k)}, \mu) &= \nabla\tilde{J}(\tilde{u}^{(k)}) - \mu\|B\tilde{u}^{(k)}\|_1 \\ &\leq -\delta\tilde{u}^{(k)T} \nabla_{\tilde{u}\tilde{u}}^2\mathcal{L}(\tilde{u}^{(k)}, \lambda^{(k)})\delta\tilde{u}^{(k)} - (\mu - \|\delta\lambda^{(k)}\|_{\infty})B\tilde{u}^{(k)},\end{aligned}\tag{2.5}$$

where  $(\delta\tilde{u}^{(k)}, \delta\lambda^{(k)})$  is the KKT point of (??), see, e.g., [?]. Therefore, if  $\nabla_{\tilde{u}\tilde{u}}^2\mathcal{L}(\tilde{u}^{(k)}, \lambda^{(k)})$  is positive definite on  $\ker(B)$ , we obtain a descend direction for  $P_1$  by the solution of (??). A globalized SQP algorithm is presented in Fig. ?? . It holds, that under sufficient conditions all accumulation points of

**Init:**  $\tilde{u}^{(0)}, \lambda^{(0)}, \varepsilon_{\text{update}} > 0, \varepsilon_{\text{tol}} > 0, \mu_0 > 0$ .

**for**  $k = 0, 1, \dots$  until convergence **do**

1. If  $\|\nabla \mathcal{L}^{(k)}\|_{\infty} \leq \varepsilon_{\text{tol}}$ , *STOP*.
2. Compute the KKT point  $(\delta \tilde{u}^{(k)}, \delta \lambda^{(k)})$  of (??).
3. Set  $\mu_{k+1} = \max\{\mu_k, \|\delta \lambda^{(k)}\|_{\infty} + \varepsilon_{\text{update}}\}$  and compute the step length  $\alpha_k$  which fulfills the Armijo condition (??) by backtracking.
4. Set  $\tilde{u}^{k+1} = \tilde{u} + \alpha_k \delta \tilde{u}^{(k)}$  and  $\lambda^{(k+1)} = \delta \lambda^{(k)}$ .

**end**

**Fig. 1:** Globalized SQP algorithm.

the sequence of iterates  $(\tilde{u}^{(k)})_{k \in \mathbb{N}}$ , generated by the algorithm in Fig. ??, are KKT points of the original problem (??), see, e.g. [?, ?, ?].

### 2.1 SQP with Quasi-Newton Methods

An important feature for the SQP approach is the fact that it does not depend on  $\nabla_{\tilde{u}\tilde{u}}^2 \mathcal{L}$  and hence the Hessian  $\nabla_{\tilde{u}\tilde{u}}^2 \mathcal{L}$  can be replaced by an approximation  $H^{(k)}$  which is symmetric and positive definite (s.p.d) on  $\ker(B)$ , in (??) and (??), without losing the theoretical properties, [?, ?, ?]. Quasi-Newton update formulas are an efficient way to calculate an approximation for the Hessian. In the context of SQP algorithms, for a given approximation  $H^{(k)}$ , we have to compute the next approximation  $H^{(k+1)}$  such that the *secant equation* holds, see, e.g., [?].

One of the most efficient Quasi-Newton update formula is the Broyden–Fletcher–Goldfarb–Shanno (BFGS) [?, ?, ?, ?] formula. An important modification for Quasi-Newton methods is the transition to limited-memory variants, where the idea is to use only the last  $m$  vectors of  $y^{(k)}$  and  $d^{(k)}$  for the new approximation.

An SQP algorithm with a Quasi-Newton approximation is outlined in Fig. ?. The KKT point of (??) is now given by the solution of

$$\begin{pmatrix} H^{(k)} & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \delta \tilde{u}^{(k)} \\ \delta \lambda^{(k)} \end{pmatrix} = - \begin{pmatrix} \nabla_{\tilde{u}} \mathcal{L}(\tilde{u}^{(k)}, \lambda^{(k)}) \\ B \tilde{u}^{(k)} \end{pmatrix}. \quad (2.7)$$

Notice, to ensure that we obtain a descent direction for  $P_1$  by the solution of (??), it is necessary that  $H^{(k)}$  is s.p.d. on  $\ker(B)$ . The BFGS update formula guarantees that  $H_{BFGS}^{(k+1)}$  is s.p.d. if  $H^{(k)}$  is s.p.d. and  $y^{(k)T} d^{(k)}$ . In unconstrained optimization, this can be ensured by the Wolfe condition [?], which is not feasible for  $P_1$ . Hence, it might happen that  $H^{(k+1)}$  is no longer

**Init:**  $\tilde{u}^{(0)}, \lambda^{(0)}, H^{(0)}, \varepsilon_{\text{update}} > 0, \varepsilon_{\text{tol}} > 0, \mu_0 > 0.$

**for**  $k = 0, 1, \dots$  **until convergence do**

1. If  $\|\nabla \mathcal{L}^{(k)}\|_{\infty} \leq \varepsilon_{\text{tol}},$  *STOP*.
2. Compute the KKT point  $(\delta\tilde{u}^{(k)}, \delta\lambda^{(k)})$  of
 
$$\min_d \nabla_{\tilde{u}} \mathcal{L}(\tilde{u}^{(k)}, \lambda^{(k)})^T d + \frac{1}{2} d^T H^{(k)} d \quad \text{s.t.} \quad B(\tilde{u}^{(k)} + d) = 0, \quad (2.6)$$
3. Set  $\mu_{k+1} = \max\{\mu_k, \|\delta\lambda^{(k)}\|_{\infty} + \varepsilon_{\text{update}}\}$  and compute the step length  $\alpha_k$  which fulfills the Armijo condition (??) by backtracking.
4. Set  $\tilde{u}^{k+1} = \tilde{u} + \alpha_k \delta\tilde{u}^{(k)}$  and  $\lambda^{(k+1)} = \delta\lambda^{(k)}$ .
5. Compute  $H^{(k+1)}$  according to some Quasi-Newton formula.

**end**

**Fig. 2:** Globalized SQP algorithm with a full Quasi-Newton approximation.

s.p.d. and the algorithm fails to compute a descent direction for  $P_1$ . In this case the whole algorithm breaks.

## 2.2 Nonlinear FETI-DP methods with Quasi-Newton

As outlined in ??, nonlinear FETI-DP can be viewed as nonlinear optimization problem with equality constraints. Thus, we use the SQP algorithm outlined in Fig. ?? to combine nonlinear FETI-DP with Quasi-Newton methods. Indeed, we solve (??) by the SQP algorithm where we approximate  $\nabla_{\tilde{u}\tilde{u}} \mathcal{L}$  using the BFGS update formula.

The efficient use of the SQP method Fig. ??, relies on the fast solution of the subproblem (??), i.e., the fast solution of the KKT system (??). Due to the block structure of  $\nabla_{\tilde{u}\tilde{u}} \mathcal{L}$  in nonlinear FETI-DP

$$\nabla_{\tilde{u}\tilde{u}} \mathcal{L}(\tilde{u}, \lambda) = \begin{pmatrix} \nabla_{\tilde{u}_B \tilde{u}_B} \mathcal{L}(\tilde{u}, \lambda) & \nabla_{\tilde{u}_B \tilde{u}_\Pi} \mathcal{L}(\tilde{u}, \lambda) \\ \nabla_{\tilde{u}_\Pi \tilde{u}_B} \mathcal{L}(\tilde{u}, \lambda) & \nabla_{\tilde{u}_\Pi \tilde{u}_\Pi} \mathcal{L}(\tilde{u}, \lambda) \end{pmatrix}, \quad (2.8)$$

where the subscript  $B$  denotes all inner and dual variables and the subscript  $\Pi$  denotes the primal variables, a factorization of  $\nabla_{\tilde{u}\tilde{u}} \mathcal{L}$  can be computed efficiently. Notice that the matrix  $\nabla_{\tilde{u}_B \tilde{u}_B} \mathcal{L}(\tilde{u}, \lambda)$  is block diagonal and local to the single subdomains. Hence, we can use  $H^{(0)} := \nabla_{\tilde{u}\tilde{u}} \mathcal{L}(\tilde{u}^{(0)}, \lambda^{(0)})$  and the property of the BFGS approximation that the application of  $H^{(k)-1}$  can be efficiently implemented, if  $H^{(0)-1}$  is known. We can solve (??) in the standard FETI-DP way by eliminating  $\delta\tilde{u}^{(k)}$  and solve the Schur complement problem

$$B H^{(k)-1} B^T \delta\lambda^{(k)} = g, \quad (2.9)$$

where

$$g = B\tilde{u}^{(k)} - B H^{(k)}{}^{-1} \nabla_{\tilde{u}} \mathcal{L}(\tilde{u}^{(k)}, \lambda^{(k)}),$$

by some Krylov method. As a preconditioner, we use the Dirichlet preconditioner evaluated at the initial iterate  $\tilde{u}^{(0)}$ . Indeed, while we update the BFGS approximation of  $\nabla_{\tilde{u}} \mathcal{L}$ , we do *not* update the Dirichlet preconditioner. In our numerical experiments, this does not increase the number of Krylov iterations for (??) significantly.

Note that it is possible to use some other initial guess for  $H^{(0)}$ , but in the context of Quasi-Newton methods with application to continuum mechanics, it is crucial to use an exact Hessian as initial value, see, e.g., [?].

To accelerate the convergence, we restart our BFGS approximation if some condition of insufficient decrease is fulfilled. Let  $(\tilde{u}^{(k)}, \lambda^{(k)})$  be some iterate and  $H^{(k)}$  be the corresponding BFGS approximation. The next iterate  $(\tilde{u}^{(k+1)}, \lambda^{(k+1)})$  is obtained by the solution of (??) and a step length  $\alpha^{(k)}$  which fulfills the Armijo condition (??). If

$$|P_1(\tilde{u}^{(k+1)}; \mu_{k+1}) - P_1(\tilde{u}^{(k)}; \mu_k)| < \eta_1 |P_1(\tilde{u}^{(k)}; \mu_k)| \quad (2.10a)$$

and

$$(1 - \eta_2) \|\nabla \mathcal{L}(\tilde{u}^{(k+1)}, \lambda^{(k+1)})\|_\infty < \|\nabla \mathcal{L}(\tilde{u}^{(k)}, \lambda^{(k)})\|_\infty, \quad (2.10b)$$

where  $0 < \eta_1, \eta_2 < 1$ , hold, we decide that the search direction, obtained with BFGS approximation  $H^{(k)}$ , neither reduces the value of the  $P_1$  nor the first order optimality condition for (??) enough. Thus, we compute the exact Hessian and set  $H^{(k+1)} := \nabla_{\tilde{u}} \mathcal{L}(\tilde{u}^{(k+1)}, \lambda^{(k+1)})$  and restart the BFGS approximation. Note, in this case we have to compute the factorization  $\nabla_{\tilde{u}} \mathcal{L}(\tilde{u}^{(k+1)}, \lambda^{(k+1)})$  and also compute the Dirichlet preconditioner for  $(\tilde{u}^{(k+1)}, \lambda^{(k+1)})$ . If (??) does *not* hold, we obtain  $H^{(k+1)}$  by a BFGS update of  $H^{(k)}$ . We outline our nonlinear FETI-DP algorithm with a globalized SQP approach and a Quasi-Newton approximation in Fig. ??.

Let us remark that there is a trade-off between the classical SQP algorithm Fig. ?? for nonlinear FETI-DP and the SQP algorithm Fig. ?? with a Quasi-Newton approximation, due to the factorization of  $\nabla_{\tilde{u}} \mathcal{L}$  within nonlinear FETI-DP. Normally, the classical SQP algorithm needs a fewer number of SQP subproblems to converge, due to the usage of the exact Hessian, but although the factorization of  $\nabla_{\tilde{u}} \mathcal{L}$  can be implemented very efficiently in parallel, there is a potential bottleneck in the (local) factorization of  $\nabla_{\tilde{u}_B \tilde{u}_B} \mathcal{L}$  and also the factorization of the Schur complement of  $S_{\Pi \Pi} = \nabla_{\tilde{u}_\Pi \tilde{u}_\Pi} \mathcal{L} - \nabla_{\tilde{u}_\Pi \tilde{u}_B} \mathcal{L} \nabla_{\tilde{u}_B \tilde{u}_B} \mathcal{L}^{-1} \nabla_{\tilde{u}_B \tilde{u}_\Pi} \mathcal{L}$  due to the runtime complexity of the sparse direct solvers. On the other hand, we can reduce the number of factorization of  $\nabla_{\tilde{u}} \mathcal{L}$  with our Quasi-Newton approach, but we need more SQP subproblems to converge and the solution of a subproblem relies on some Krylov method. Hence, we increase the total amount of Krylov iterations, which also increases the communication in a parallel implementation.

```

Init:  $\tilde{u}^{(0)}, \lambda^{(0)}, \varepsilon_{\text{update}} > 0, \varepsilon_{\text{tol}} > 0, \mu_0 > 0, \eta_1 \in (0, 1), \eta_2 \in (0, 1),$ 
 $H^{(0)} := \nabla_{\tilde{u}\tilde{u}}\mathcal{L}(\tilde{u}^{(0)}, \lambda^{(0)})$ , compute a factorization of  $H^{(0)}$ .
for  $k = 0, 1, \dots$  until convergence do
  1. If  $\|\nabla\mathcal{L}^{(k)}\|_{\infty} \leq \varepsilon_{\text{tol}}$ , STOP.
  2. Compute the KKT point  $(\delta\tilde{u}^{(k)}, \delta\lambda^{(k)})$  of
      
$$\min_d \nabla_{\tilde{u}}\mathcal{L}(\tilde{u}^{(k)}, \lambda^{(k)})^T d + \frac{1}{2}d^T H^{(k)}d \quad \text{s.t.} \quad B(\tilde{u}^{(k)} + d) = 0, \quad (2.11)$$

  3. Set  $\mu_{k+1} = \max\{\mu_k, \|\delta\lambda^{(k)}\|_{\infty} + \varepsilon_{\text{update}}\}$  and compute the step length  $\alpha_k$ 
      which fulfills the Armijo condition (??) by backtracking.
  4. Set  $\tilde{u}^{k+1} = \tilde{u} + \alpha_k \delta\tilde{u}^{(k)}$  and  $\lambda^{(k+1)} = \delta\lambda^{(k)}$ .
  5. If (??) holds, then
      Set  $H^{(k+1)} := \nabla_{\tilde{u}\tilde{u}}\mathcal{L}(\tilde{u}^{(k+1)}, \lambda^{(k+1)})$ 
      and compute a factorization of  $H^{(k+1)}$ 
    else
      Compute  $H^{(k+1)}$  by the BFGS update.
end

```

**Fig. 3:** Nonlinear FETI-DP with a globalized SQP algorithm and a Quasi-Newton approximation.

### 3 Numerical Results

We consider a two- or three-dimensional beam bending benchmark problem with a compressible Neo-Hookean constitutive law using  $Q2$  finite elements. The strain energy density function is given by  $J(x) = \frac{\mu}{2}(\text{tr}(F(x)^T F(x)) - 3) - \mu \log(\psi(x)) + \frac{\lambda}{2}(\log(\psi(x)))^2$ , where  $\psi(x) = \det(F(x))$ ,  $F(x) = \nabla\varphi(x)$ ,  $\varphi(x) = x + u(x)$ ,  $u(x)$  denotes the displacement and  $\mu$  and  $\lambda$  are the Lamé constants. As material parameters, we use  $E = 210$  and  $\nu = 0.3$ .

As a baseline, we use a globalized Newton-like algorithm based on the exact differentiable penalty function  $P(\tilde{u}, \lambda; \mu) := \mathcal{L}(\tilde{u}, \lambda) + \frac{\mu}{2}\|B\tilde{u}\|^2 + \|B\nabla_{\tilde{u}}\mathcal{L}(\tilde{u}, \lambda)\|$  introduced by Di Pillo and Grippo in [?]. Note that a Newton-like direction for  $P$  can be computed by the solution of the standard Lagrange-Newton equation. For the combination of nonlinear FETI-DP methods and the penalty function  $P$  see, e.g., [?, ?].

In Table ??, we show weak scaling results for the two-dimensional problem comparing our SQP approach using Quasi-Newton approximation of the Hessian with the base line using the exact differentiable penalty function  $P$ . We increase the number of subdomains from 40 up to 640 using subdomains with 206 082 degrees of freedom. We report the nonlinear solution time, the number of (nonlinear) iterations, and the total number of Krylov iterations. Note that for both methods one nonlinear iteration involves the solution of the saddle point problem (??), where we use a Quasi-Newton approximation for the SQP approach and the exact Hessian for the penalty function  $P$ . But

**Table 1:** Comparison of Quasi-Newton SQP approach with a Newton-like line search method for  $P$  in 2D, local subdomains =  $160 \times 160$ , #dofs subdomain = 206 082

Sizes for local subdomains of $160 \times 160$ $Q_2$ -elements							
#Subds	#Dofs $(\bar{x}, \lambda)$	#Dofs $(\bar{x})$	#Coarse Dofs				
40	8 280 168	8 242 856	348				
160	33 148 812	32 971 152	1 500				
360	74 605 936	74 184 888	3 452				
640	132 651 540	131 884 064	6 204				

  

#Subds	Quasi-Newton SQP				Newton-like method for $P$		
	Solve	#It.	#Krylov It.	#Recomp ( $\nabla_{\bar{u}\bar{u}}^2 \mathcal{L}$ )	Solve	#Newton It.	#Krylov It.
40	321.1s	25	589	8	400.7s	15	355
160	421.0s	26	624	8	514.8s	14	341
360	381.0s	24	578	7	522.8s	14	349
640	422.4s	25	615	8	525.7s	14	355

**Table 2:** Comparison of Quasi-Newton SQP approach with a Newton-like line search method for  $P$  in 3D, local subdomains =  $18 \times 18 \times 18$ , #dofs subdomain = 151 959

Sizes for local subdomains of $18 \times 18 \times 18$ $Q_2$ -elements							
#Subds	#Dofs $(\bar{x}, \lambda)$	#Dofs $(\bar{x})$	#Coarse Dofs				
80	12 815 196	12 152 964	1 800				
640	104 010 420	97 213 008	16 020				

  

#Subds	Quasi-Newton SQP				Newton-like method for $P$		
	Solve	#It.	#Krylov It.	#Recomp ( $\nabla_{\bar{u}\bar{u}}^2 \mathcal{L}$ )	Solve	#Newton It.	#Krylov It.
80	939.0s	22	1 415	5	2 010.9s	14	948
640	486.9s	11	596	2	1 818.1s	10	857

for the Newton-like method based on  $P$ , we need to compute a factorization of  $\nabla_{\bar{u}\bar{u}}^2 \mathcal{L}$  in each nonlinear iteration which is not the case for the SQP approach using Quasi-Newton approximations, where we only need to compute such a factorization if we recompute the original Hessian of  $\mathcal{L}$  as outlined in Fig. ???. Hence, the computational saving is in the fewer factorizations of  $\nabla_{\bar{u}\bar{u}}^2 \mathcal{L}$ . We need between 7 and 8 factorizations for the SQP approach, whereas we need between 14 and 15 factorizations for  $P$ , which means we can save nearly 50% of the factorizations. Nevertheless, we need more (nonlinear) iterations for the SQP approach, between 24 and 26, compared to the 14 or 15 iterations for  $P$ , resulting in a higher number of (total) Krylov iterations for the SQP approach. The higher number of Krylov iterations consumes some of the time savings. Overall, we obtain a speed up of the solution between 18% for 160 subdomains and 27% for 360 subdomains for the Quasi-Newton method using SQP approach against our base line based on  $P$ .

In Table ??, we show weak scaling results for the three-dimensional system. Due to the structured decomposition in our experiments, we only report

results for 80 and 640 subdomains. The number of factorizations of  $\nabla_{\bar{u}\bar{u}}^2 \mathcal{L}$  in the SQP approach reduces drastically from 14 and 10 for the penalty method based on  $P$  to 5 and 2, for 80 and 640 subdomains, respectively. Therefore, also the nonlinear solution times reduces over 50%. This shows that Quasi-Newton methods can accelerate the solution time by reducing the number of factorizations of  $\nabla_{\bar{u}\bar{u}}^2 \mathcal{L}$ .

Note that using a Quasi-Newton method for  $P$  is very challenging, since  $\nabla^2 \mathcal{L}$  is not a Newton-like approximation for  $\nabla^2 P$  only the solution of the standard Lagrange-Newton equation reveals a Newton-like search direction. Furthermore, the recomputation of  $\nabla_{\bar{u}\bar{u}}^2 \mathcal{L}$  is essential to obtain a good speed up, i.e., without the recomputation the SQP approach with Quasi-Newton approximations is slower than the Newton-like method for  $P$ .

## References

1. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific J. Math.* **16**(1), 1–3 (1966). DOI 10.2140/pjm.1966.16.1
2. Barnafi, N.A., Pavarino, L.F., Scacchi, S.: Parallel inexact Newton–Krylov and Quasi–Newton solvers for nonlinear elasticity. *Computer Methods in Applied Mechanics and Engineering* **400**, 115557 (2022). DOI 10.1016/j.cma.2022.115557
3. 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)
4. Broyden, C.G.: The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA J. Appl. Math.* **6**(1), 76–90 (1970). DOI 10.1093/imamat/F6.1.76
5. Di Pillo, G., Grippo, L.: A new class of augmented Lagrangians in nonlinear programming. *SIAM J. Control Optim.* **17**(5), 618–628 (1979). DOI 10.1137/0317044
6. Fletcher, R.: A new approach to variable metric algorithms. *Comput. J.* **13**(3), 317–322 (1970). DOI 10.1093/comjnl/13.3.317
7. Goldfarb, D.: A family of variable-metric methods derived by variational means. *Math. Comp.* **24**(109), 23–26 (1970). DOI 10.1090/S0025-5718-1970-0258249-6
8. Klawonn, A., Lanser, M., Rheinbach, O., Uran, M.: Nonlinear FETI–DP and BDDC methods: a unified framework and parallel results. *SIAM J. Sci. Comput.* **39**(6), C417–C451 (2017). DOI 10.1137/16M1102495
9. Köhler, S.: *Globalization of Nonlinear FETI–DP Methods*. Ph.D. thesis, Technische Universität Bergakademie Freiberg (2024)
10. Köhler, S., Rheinbach, O.: Globalization of Nonlinear FETI–DP Domain Decomposition Methods Using an SQP Approach. *Vietnam J. Math.* pp. 1–27 (2022). DOI 10.1007/s10013-022-00567-2
11. Köhler, S., Rheinbach, O.: Globalization of Nonlinear FETI–DP Methods. In: *Domain Decomposition Methods in Science and Engineering XXVI, Lect. Notes Comput. Sci. Eng.*, vol. 145, pp. 327–334. Springer Cham (2022). DOI 10.1007/978-3-030-95025-5\_36. URL <http://www.ddm.org/DD26/proceedings/327.pdf>
12. Nocedal, J., Wright, S.: *Numerical Optimization*. Springer Science & Business Media (2006). DOI 10.1007/978-0-387-40065-5
13. Shanno, D.F.: Conditioning of Quasi-Newton Methods for Function Minimization. *Math. Comp.* **24**(111), 647–656 (1970). DOI 10.1090/S0025-5718-1970-0274029-X

14. Toselli, A., Widlund, O.: Domain decomposition methods—algorithms and theory, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin (2005)
15. Ulbrich, M., Ulbrich, S.: Nichtlineare Optimierung. Birkhäuser Basel (2012). DOI 10.1007/978-3-0346-0654-7
16. Wolfe, P.: Convergence conditions for ascent methods. *SIAM review* **11**(2), 226–235 (1969). DOI 10.1137/1011036