



Domain decomposition for 3D nonlinear magnetostatic problems: Newton-Krylov-Schur vs. Schur-Newton-Krylov methods

Mohamed Ghenai, Ronan Perrussel, Olivier Chadebec, Frédéric Vi,
Jean-Michel Guichon, Gérard Meunier, Jonathan Siau

► To cite this version:

Mohamed Ghenai, Ronan Perrussel, Olivier Chadebec, Frédéric Vi, Jean-Michel Guichon, et al.. Domain decomposition for 3D nonlinear magnetostatic problems: Newton-Krylov-Schur vs. Schur-Newton-Krylov methods. IEEE Transactions on Magnetism, 2024, 60 (3), 10.1109/TMAG.2023.3299989 . hal-04183878

HAL Id: hal-04183878

<https://hal.science/hal-04183878>

Submitted on 21 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Domain decomposition for 3D nonlinear magnetostatic problems: Newton-Krylov-Schur vs. Schur-Newton-Krylov methods

Mohamed I. Ghenai^{1,2,3}, Ronan Perrussel², Olivier Chadebec¹, Frédéric Vi³,
Jean-Michel Guichon¹, Gérard Meunier¹, and Jonathan Siau³

¹Univ. Grenoble Alpes, CNRS, Grenoble INP, G2ELab, F-38000, Grenoble, France

²LAPLACE, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France

³Altair Engineering, Meylan, France

Domain decomposition is a strategy designed to be used on parallel machines. This strategy leads to hybrid methods between direct and iterative solvers and allows users to benefit from the advantages of both. Lately, the growing size of simulations in electromagnetics brought to light the interest of using domain decomposition. Nonlinearity is also one of the problems specificities where the need for an efficient solver is high. This paper provides a comparison between two techniques of domain decomposition for solving 3D nonlinear magnetostatic problems. A contactor test case with two nonlinear materials was used to estimate the performances of both methods.

Index Terms—Domain decomposition, FETI method, Nonlinear magnetostatic, Parallel computing.

I. INTRODUCTION

DOMAIN decomposition is a strategic approach that aims to decompose complex problems into smaller subdomains. This decomposition enables an independent processing of subdomains, leveraging parallelization for efficient computations. Each subdomain is equipped with its own local system, employing tailored solving methods that cater to specific properties like size and matrix characteristics.

Although domain decomposition is a popular approach in various fields such as fluid dynamics and structural mechanics, it is not commonly used in electromagnetic simulations. In electromagnetic simulations, the fields are interconnected and can propagate throughout the entire computational domain. This makes it challenging to decompose the domain into smaller subdomains without introducing errors at the boundaries. A highly refined mesh is usually needed in electromagnetic simulation which add an other specification to be taken into account when using the domain decomposition.

In the context of nonlinear problems, it is often observed that the nonlinearity is localized rather than spread across the entire system [1]. For example, in magnetostatic simulations, a significant portion of the finite element discretization volume represents air, while regions with nonlinear materials are confined to smaller areas. Exploiting this characteristic, the domain decomposition method efficiently distributes the workload associated with nonlinearity among subdomains. Consequently, there is no need to calculate the tangent operator at each Newton-Raphson step in linear subdomains. Instead, local nonlinear solutions can be performed on smaller, more manageable problems, replacing the need to solve a much larger global problem. Moreover, this approach allows for the execution of the complete simulation in parallel, further enhancing efficiency.

Another valuable application of domain decomposition is accelerating the global nonlinear solving process. By dividing the workload after the linearization step, domain decomposition

effectively reduces computational burden, leading to faster solutions. Over time, domain decomposition has proven to be an extremely powerful and high-performance method for solving large-scale problems in parallel [2].

The significance of this paper lies in its contribution of a detailed comparison between two non-overlapping domain decomposition techniques originally developed for nonlinear mechanics simulations [1]. The paper presents a comprehensive overview of the implementation of these methods and illustrates their practical application in a specific scenario involving a 3D contactor with nonlinear ferromagnetic materials.

II. PROBLEM DESCRIPTION

A 3D magnetostatic problem using a scalar potential formulation will be considered in the reminder of this paper [3]. Using the finite element method, for a domain Ω with a magnet as source, ferromagnetic materials and the device surrounded by air, the equations will be

$$\text{div}[\mu(H)(-\text{grad}(\phi) + T)] = 0, \quad (1)$$

where ϕ is the magnetic scalar potential and T the source term. For nonlinear materials μ is a function of the magnetic field $H = -\text{grad}(\phi) + T$, the system to be solved is written as

$$K(u)u = b, \quad (2)$$

where

- $K(u)$ is the finite element matrix of the magnetic scalar potential formulation,
- u contains the values of the magnetic scalar potential at each node of the mesh,
- b is the source vector.

The Newton-Raphson iterative method is usually considered to solve (2). This method consists of computing the tangent matrix DK as well as the residual r_u at every iteration n and solving successive linear systems of the form

$$DK(u^n)\delta u^n = -r_u(u^n). \quad (3)$$

The residual is given by $r_u(u^n) = K(u^n)u^n - b(u^n)$.

III. DOMAIN DECOMPOSITION

Domain decomposition is used to solve partial differential equations. It is considered as a hybrid method between robust yet computationally expensive direct solvers and naturally parallel but less robust iterative solvers. This approach involves dividing a given domain, denoted as Ω , into multiple smaller subdomains, represented as Ω_i , and orchestrating the solution process across these subdomains. Within the domain decomposition framework, two primary types of methods emerge: those with overlap, commonly referred to as Schwartz methods, and those without overlap, recognized as Schur methods.

For the purpose of this study, our focus centers specifically on non-overlapping domain decomposition methods. While overlapping methods use nodes beyond the interface between subdomains (called ghost nodes), non-overlapping methods are characterized by the absence of shared surfaces (in the context of two-dimensional systems) or shared volumes (in the context of three-dimensional systems) connecting neighboring subdomains [4]. The selection of non-overlapping methods was motivated by the parallel computing capabilities that it offers. The absence of shared sections is more intuitive for having independent subdomains that can be treated separately. Overlapping methods need synchronization on the ghost nodes which means more communication to ensure the solution continuity.

A. FETI : Finite Element Tearing and Interconnecting

This method was introduced by Farhat and Roux in 1991 [5]. The domain is decomposed into subdomains and each subdomain is processed independently on a processor. The subdomain problems are usually solved by a direct solver, while an iterative method enables to solve a global problem condensed on the interface between subdomains. With the use of an appropriate preconditionner (the Dirichlet or the lumped preconditioner), it has been shown that the FETI method is optimal for a scalar diffusion problem [2]. For this article only the case with a conforming mesh is considered.

The method used for our magnetostatic problem guarantees the continuity of both normal and tangent components of the magnetic field.

If the domain Ω is decomposed into N_{sd} subdomains, on a subdomain s , a Poisson type problem has to be solved

$$K_s u_s = b_s + B_s^t \lambda, \quad (4)$$

where K_s , u_s , and b_s are respectively matrix, solution and right-hand-side relative to subdomain s , B_s is the signed Boolean matrix that extract from the vector u_s its signed restriction to the subdomain interface shared with its neighbors (opposite signs between two neighboring subdomains) and λ is the Lagrange multiplier vector required to enforce the continuity conditions (5) for all couples of neighbors as,

$$\sum_{s=1}^{N_{sd}} B_s u_s = 0. \quad (5)$$

Floating subdomains need also to be defined. They are subdomains which do not intersect the Dirichlet boundary: the pseudo inverse matrix K^+ is then used instead of the inverse K^{-1} . The bases of the kernel of each matrix K_s are concatenated in the rigid body mode matrix R [5]. For a scalar potential problem, the kernel of K_s is at most of dimension 1. R^i is the restriction of R on the interface. Combining the two previous equations with an orthogonality condition [5] gives the problem condensed on the global interface between subdomains that can be written as

$$\begin{pmatrix} F^i & R^i \\ (R^i)^t & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} -\sum_{s=1}^{N_{sd}} B_s K_s^+ b_s \\ R^t b \end{pmatrix}, \quad (6)$$

where α is a vector of size of the number of floating subdomains and F^i the global matrix of the problem condensed on the interface

$$F^i = \sum_{s=1}^{N_{sd}} B_s K_s^+ B_s^t. \quad (7)$$

The interface problem is solved using the preconditioned conjugate projected gradient (PCPG) algorithm [1].

Algorithm 1 PCPG

Require: $\lambda^0 = R^i (R^{it} R^i)^{-1} R^t f$, $r^k = (\sum_{s=1}^{s=N} B_s K_s^+ f_s) - F^i \lambda^0$

while $r^t Pr \leq tol$ **do**

$w^{k-1} \leftarrow [I - R^i (R^{it} R^i)^{-1} R^{it}] r^{k-1}$ ▷ Projecting

$z^{k-1} \leftarrow P w^{k-1}$ ▷ Preconditioning

$y^{k-1} \leftarrow [I - R^i (R^{it} R^i)^{-1} R^{it}] z^{k-1}$ ▷ Projecting

$\gamma^k \leftarrow y^{k-1} w^{k-1} / y^{k-2} w^{k-2}$ ▷ $\gamma^1 \leftarrow 0$

$s^k \leftarrow y^{k-1} + \gamma^k s^{k-1}$ ▷ $s^1 \leftarrow 0$

$\beta^k \leftarrow y^{k-1} w^{k-1} / s^{k-1} F^i s^k$

$\lambda^k \leftarrow \lambda^{k-1} + \beta^k s^k$

$r^k \leftarrow r^{k-1} - \beta^k F^i s^k$

$k \leftarrow k + 1$

end while

At each iteration, one local solve by subdomain is executed. The local matrix is factorized and stored once and used to compute the new vector. In a multiprocessors implementation, these local solutions are fully independent and can be executed in parallel. Only one global communication to compute the global reduction sum is needed.

After computing λ , α can be evaluated by

$$\alpha = ((R^i)^t R^i)^{-1} (R^i)^t (-F^i \lambda + \sum_{i=1}^{N_{sd}} \sum_{j=1}^{a_j} (B K_i^+ f_i - B K_j^+ f_j)). \quad (8)$$

Two well-known preconditioners can be considered for the FETI method: the Dirichlet preconditioner and the lumped preconditioner. The Dirichlet preconditioner is given by

$$D_I^{-1} = \sum_{s=1}^{N_j} B^s \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^s - K_{ib}^{s^T} K_{ii}^{s-1} K_{ib}^s \end{bmatrix} B^{s^T} \quad (9)$$

with B^s the jump operator from global interface to local interface.

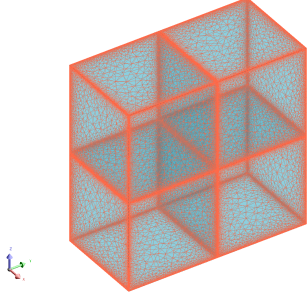


Fig. 1. Linear magnet test case decomposed into 4 subdomains.

The lumped preconditioner simplifies the construction of the preconditioner by aggregating off-diagonal terms of the stiffness matrix onto the diagonal through a lumping process. This results in a simplified and localized preconditioner that can be efficiently inverted. However, it may be less effective compared to more sophisticated preconditioning techniques in scenarios with highly varying subdomain sizes or heterogeneous materials.

The results on the linear test cases illustrated in Fig. 1 (linear magnet with residual induction $B_r = 0.85 T$ on axis x , decomposed into 4 subdomains) showed that if the scenario where, 1/ one subdomain is solved by one processor, 2/ the size of the subdomain corresponds to the MUMPS [7] and PETSc [8] recommendations and 3/ the workload is well balanced, then the scalability results showed a linear performance upgrade.

The results in terms of speedup for 105k nodes are presented in Table I and Fig. 2.

TABLE I
RESULTS COMPARISON FOR THE MAGNET TEST CASE

Item	Standard	DDM
Time in sec	7.61	2.24
Speedup	/	3.4

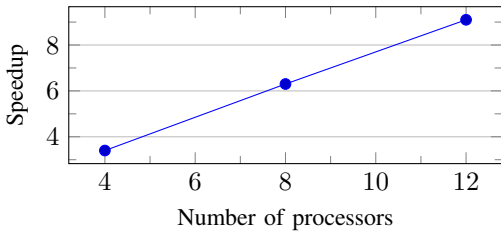


Fig. 2. Scalability plot for the linear case.

B. Newton-Krylov-Schur

The Newton-Krylov-Schur (NKS) method can be implemented in a straightforward manner. During each Newton-Raphson iteration, the FETI method is employed to solve the tangent linear system (3). Similar to FETI, the Schur complement within the NKS framework characterizes the interface

problem. Importantly, the NKS methods do not require the assembly of the global problem, and the computation of the residual is partitioned across the subdomains [9] [1].

By adopting this approach, the NKS methods effectively utilize the FETI method to solve the tangent linear system at each Newton-Raphson iteration. This circumvents the need for assembling the global problem, resulting in improved computational efficiency. Furthermore, by partitioning the residual computation across the subdomains, parallel processing can be leveraged to enhance performance and reduce computational time. The NKS method algorithm is summarized in [2]

Algorithm 2 NKS with FETI algorithm

Require: $p = 1; u^1 = 0$ ▷ Solve Global NL problems
while $\delta u \leq tol_{NR}$ **do**
 Solve $DK(u^p)\delta^p u = -r_u^p(u^p)$ with PCPG
 $u^{p+1} \leftarrow u^p + \delta^p u$
 $p \leftarrow p + 1$
end while

C. Schur-Newton-Krylov

The Schur-Newton-Krylov (SNK) methods were developed to address a crucial observation: the existing NKS methods do not fully leverage the potential of domain decomposition when solving nonlinear problems. While NKS methods primarily utilize domain decomposition for solving tangent linear systems, SNK methods take a step further. In SNK methods, the global nonlinear problem undergoes decomposition into smaller, localized nonlinear problems, accompanied by the imposition of continuity conditions on the interface unknowns across subdomains [1]. This decomposition process gives the following linearized condensed system :

$$\begin{pmatrix} DF^i & R^i \\ (R^i)^t & 0 \end{pmatrix} \begin{pmatrix} \delta\lambda \\ \delta\alpha \end{pmatrix} = \begin{pmatrix} -r_\lambda \\ 0 \end{pmatrix}. \quad (10)$$

Problem (10), as (7), is solved using the PCPG as in Algorithm 1. The SNK method is summarized in Algorithm [3].

IV. APPLICATION

Both NKS and SNK algorithms have been implemented in Altair FluxTM software [10]. They have been tested and compared to the standard Newton-Raphson method commonly used for this type of simulations. The linear solver used in Newton Raphson by default is a direct solver (MUMPS). The test case is illustrated in Fig. 3 and represents a 3D magnetic contactor. It was decomposed into 4 subdomains: 1 air (linear), 2 upper pallet (nonlinear), 3 magnet (linear) and 4 E shape (nonlinear). The size of the test case is 2432 nodes.

TABLE II
RESULTS COMPARISON FOR THE CONTACTOR TEST CASE

Item	NR	NKS	SNK
Max local NL iterations	/	/	5
Global NL iterations	6	6	7
Speedup	1	1.3	0.4

Algorithm 3 SNK with FETI algorithm

Require: $n = 0, \lambda^0 = R^i(R^{it}R^i)^{-1}R^tb$
 $P = I - R^i(R^{it}R^i)^{-1}R^{it}$
 $K_s(u_s^n) = b_s(u_s^n) - B_s^t\lambda_s^n$
while $\delta u \leq \text{tol}_{NR}$ **do**
 Solve $DK_s(u_s^{n^p})\delta^p u = b_s^p(u_s^{n^p}) - B_s^t\lambda_s^n - K_s(u_s^{n^p})$
 $u_s^{n^{p+1}} \leftarrow u_s^{n^p} + \delta^p u$
 $p \leftarrow p + 1$
end while
 $r_\lambda^n \leftarrow \sum_{s=1}^{N_{sd}} B_s u_s^n$ \triangleright Compute residual
while $\|\delta\lambda^n\|/\|\lambda^n\| \leq \text{tol}_g$ **do**
 Solve FETI interface problem with PCPG
 $\lambda^{n+1} \leftarrow \lambda^n + P\delta\lambda^n$
 $\alpha^{n+1} \leftarrow \alpha^n + ((R^i)^t R^i)^{-1}(R^i)^t(-DF^i(\lambda^n)\delta\lambda^n - r_\lambda^n)$
 $u_s^{n+1} \leftarrow u_s^n - DK_s^{-1}B_s^t P\delta\lambda^n - (R^i)^t\delta\alpha_s^n$
 $n \leftarrow n + 1$
 $K_s(u_s^n) = b_s(u_s^n) - B_s^t\lambda_s^n$
 $p = 1; u_s^{n^1} = u_s^n$
 while $\delta u \leq \text{tol}_{NR}$ **do**
 Solve $DK_s(u_s^{n^p})\delta^p u = b_s^p(u_s^{n^p}) - B_s^t\lambda_s^n - K_s(u_s^{n^p})$
 $u_s^{n^{p+1}} \leftarrow u_s^{n^p} + \delta^p u$
 $p \leftarrow p + 1$
 end while
 $r_\lambda^n \leftarrow \sum_{s=1}^{N_{sd}} B_s u_s^n$ \triangleright Compute residual
end while

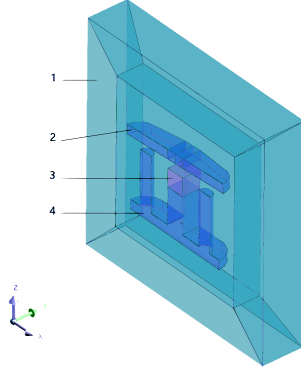


Fig. 3. Contactor test case with nonlinear materials.

Table [II] sums up the results. An adaptive convergence criteria has been used to obtain less PCPG iterations with a Dirichlet preconditioner [2]. The problem was solved using 4 cores. For domain decomposition this means that we solve one subdomain per processor which is the optimal situation when workload balancing is good. The max local nonlinear iterations number is the upper limit of local nonlinear solving of the two nonlinear subdomains. The speedup is the fraction between global solving time of the reference standard Newton Raphson method and new domain decomposition methods. This include assembly and solving times. The tolerance on the relative error is of 10^{-5} .

VI. ANALYSIS

The test case reveals that the SNK method exhibits a high number of local nonlinear iterations, resulting in a significant

convergence delay within the global loop. Consequently, the performance of this method is adversely affected, leading to noticeable slowdown. In contrast, the NKS method demonstrates favorable performance when compared to the standard Newton Raphson method.

Furthermore, employing the domain decomposition method as a linear solver yields promising results, with a 30% speedup observed. This outcome aligns with previous findings regarding the linear case, further validating the effectiveness of the domain decomposition approach. It should be noted that the workload between subdomains in this specific nonlinear test case was not well balanced, as subdomain “air” concentrates a large portion of the global mesh. Partitioning large physic regions will certainly improve balancing to obtain comparable performances as in linear case.

VI. CONCLUSION

Domain decomposition offers a compelling alternative to standard solvers, effectively accelerating simulations and improving overall performances. In the context of nonlinear simulations, a comparison was conducted between two methods. The results exhibited favorable performances and a significant speedup. Notably, the NKS method demonstrated a relative ease of implementation, further augmenting its benefits. Conversely, the SNK method necessitated a large number of iterations to achieve convergence, negatively impacting the overall performance. However, it is worth considering that switching from the FETI-1 method to FETI-Dual Primal may potentially enhance the convergence behavior of the SNK method on this first test case [11].

REFERENCES

- [1] C. Negrello, P. Gosselet, and C. Rey, “Nonlinearly preconditioned FETI solver for substructured formulations of nonlinear problems,” *Mathematics* 9.24 (2021), p. 3165.
- [2] C. Farhat, J. Mandel, and F. X. Roux, “Optimal convergence properties of the FETI domain decomposition method,” *Computer methods in applied mechanics and engineering* 115 (1994), pp. 365–385.
- [3] G. Meunier, ed., “The finite element method for electromagnetic modeling,” John Wiley & Sons, 2010, pp. 69-83.
- [4] N. Spillane, “Robust domain decomposition methods for symmetric positive definite problems,” PhD thesis. Paris 6, 2014.
- [5] C. Farhat, and F. X. Roux, “A method of finite element tearing and interconnecting and its parallel solution algorithm,” *International journal for numerical methods in engineering* 32 (1991), pp. 1205–1227.
- [6] A. Klawonn, and O. Widlund, “FETI and Neumann-Neumann iterative substructuring methods: connections and new results,” *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences* 54.1 (2001): 57-90.
- [7] P. R. Amestoy, I. S. Duff, J. Koster and J. Y. L’Excellent, “A fully asynchronous multifrontal solver using distributed dynamic scheduling,” *SIAM Journal on Matrix Analysis and Applications*, Vol 23, No 1, pp 15-41 (2001).
- [8] Balay, S., Gropp, W., McInnes, L. C. and Smith, B. F. (1998). PETSc, the portable, extensible toolkit for scientific computation. Argonne National Laboratory, 2(17).
- [9] J. Pebrel, C. Rey, and P. Gosselet, “A nonlinear dual-domain decomposition method: Application to structural problems with damage,” *International Journal for Multiscale Computational Engineering* 6.3 (2008).
- [10] Altair Engineering, “Altair Flux™,” <https://www.altair.com/flux>.
- [11] C. Negrello, P. Gosselet, C. Rey, and J. Pebrel “Substructured formulations of nonlinear structure problems - influence of the interface condition,” *International Journal for Numerical Methods in Engineering*, 107(13), 1083-1105 (2016).