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

Mohamed I. Ghenai\textsuperscript{1,2,3}, Ronan Perrussel\textsuperscript{2}, Olivier Chadebec\textsuperscript{1}, Frédéric Vi\textsuperscript{3}, Jean-Michel Guichon\textsuperscript{1}, Gérard Meunier\textsuperscript{1}, and Jonathan Siau\textsuperscript{3}

\textsuperscript{1}Univ. Grenoble Alpes, CNRS, Grenoble INP, G2ELab, F-38000, Grenoble, France
\textsuperscript{2}LAPLACE, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France
\textsuperscript{3}Altair 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 test case illustrates the results that can be expected.

\textbf{Index Terms}—Domain decomposition, FETI method, Nonlinear magnetostatic, Parallel computing.

\section{Introduction}

Domain decomposition is a strategy that seeks to decompose the problem into subdomains in order to be able to process them independently by relying on parallelization. Each subdomain will thus have its own local system to solve and will rely on solving methods adapted to its specificities (size, matrix properties...). For the case of nonlinear problems, most of the time, the nonlinearity is localized. This enables the domain decomposition method to distribute the workload resulting from the nonlinearity on some subdomains and have linear subdomains, where no need to calculate the tangent operator at each Newton-Raphson step. Hence, a local nonlinear solution can be performed on smaller problems instead of solving a much bigger global problem, while executing the full simulation in parallel. The originality of this paper is to provide a comparison between two non-overlapping domain decomposition techniques developed initially to treat nonlinear mechanics simulations. This paper details the implementation of these methods and shows an application of its use on a 3D contactor with nonlinear ferromagnetic materials.

\section{Problem Description}

A 3D magnetostatic problem using a scalar potential formulation will be considered in the reminder of this paper \cite{1}. Using the finite element method, for a domain $\Omega$ with a magnet as source, ferromagnetic materials and the device surrounded by air, the system to be solved is written as

$$K(u)u = b(u), \quad (1)$$

where

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

The Newton-Raphson iterative method is usually considered to solve \cite{1}. 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

$$DK(u^n)\delta u^n = -r_u(u^n). \quad (2)$$

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

\section{Domain Decomposition}

For this study, we consider non-overlapping domain decomposition methods. They are characterized by the absence of a shared surface (2D) or volume (3D) connecting neighboring subdomains \cite{2}. This choice was motivated by the parallel computing capabilities that it offers.

\subsection{A. FETI : Finite Element Tearing and Inteconnecting}

This method was introduced by Farhat and Roux in 1991 \cite{3}. 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 a proper preconditioner, it has been shown that the FETI method is optimal for a scalar diffusion problem \cite{4}.

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

$$K_s u_s = b_s + B^*_s \lambda, \quad (3)$$

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 $\lambda$ is the Lagrange multiplier vector required to enforce the continuity conditions \cite{4} for all couples of neighbors,

$$\sum_{s=1}^{N_{sd}} B_s u_s = 0. \quad (4)$$

\textsuperscript{1}Mohamed I. Ghenai, \textsuperscript{2}Ronan Perrussel, \textsuperscript{3}Olivier Chadebec, Frédéric Vi, Jean-Michel Guichon, Gérard Meunier, and Jonathan Siau

\textsuperscript{1}Univ. Grenoble Alpes, CNRS, Grenoble INP, G2ELab, F-38000, Grenoble, France
\textsuperscript{2}LAPLACE, Université de Toulouse, CNRS, INPT, UPS, Toulouse, France
\textsuperscript{3}Altair Engineering, Meylan, France
Floating subdomains need also to be defined. They are subdomains without Dirichlet-type boundary conditions: 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 $R$. For a scalar problem the kernel of $K_s$ is at most of dimension 1. $R^i$ is the restriction of $R$ on the interface. The problem condensed on the global interface between subdomains can then be written as

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

(5)

where $\alpha$ is of size the number of floating subdomains and

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

(6)

B. Newton-Krylov-Schur methods

The Newton-Krylov-Schur (NKS) methods can be implemented straightforwardly: at each Newton-Raphson iteration, the FETI method is considered to solve the tangent linear system (2). As in FETI, the Schur complement in NKS describes the interface problem. The global problem is never assembled and the residual computation is partitioned on the subdomains [5].

C. Schur-Newton-Krylov

The Schur-Newton-Krylov (SNK) methods were developed based on the observation that NKS methods do not take advantage of domain decomposition to solve the nonlinear problem, but only to solve the resulting tangent linear systems. In SNK methods the global nonlinear problem is decomposed into small nonlinear problems with continuity conditions imposed on the interface unknowns across the subdomains, resulting in a condensed nonlinear interface problem with tangent system

$$
\begin{pmatrix}
DF^i \\
(R^i)^t \\
0
\end{pmatrix}
\begin{pmatrix}
\delta \lambda \\
\delta \alpha
\end{pmatrix}
= 
\begin{pmatrix}
- r^i_p \\
0
\end{pmatrix}.
$$

(7)

Problem (7), just as (6), is solved using a preconditioned conjugate projected gradient (PCPG) algorithm as in [3]. The algorithm of SNK methods is summarized in Algorithm 1.

Algorithm 1 SNK with FETI algorithm

Require: $n = 1$, $\lambda^0 = R^t (R^t R^t)^{-1} R^t b$

while $\|\delta \lambda^n\|/\|\lambda^n\| \leq tol_R$ do

\quad $K_s(u^n_s) = b_s(u^n_s) - B_s^t \lambda^n_s$ \quad $\triangleright$ Solve local NL problems

\quad $p = 1$; $u^{n+1}_s = (R^t)^t \alpha^n_s$

while $\|\delta u^n\| \leq tol_{NR}$ do

\quad Solve $DK_s(u^n_s) \delta u^n = b^n_s(u^n_s) - B_s^t \lambda^n_s$

\quad $u^{n+1}_s \leftarrow u^n_s + \delta u^n$

\quad $p \leftarrow p + 1$

end while

end while

IV. Application

Both NKS and SNK algorithms have been implemented in Altair Flux™ software [6]. 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 (LU factorisation). The test case is illustrated in Fig. 1 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 I sums up the results. An adaptive convergence criteria has been used to obtain less PCPG iterations with a Dirichlet preconditioner [4]. The project was solved using 4 cores.

REFERENCES


