Mesh decomposition for efficient parallel computing of electrical machines by means of FEM accounting for motion

Stefan Böhmer[1], Enno Lange[1], Martin Hafner[1], Tim Cramer[2], Christian Bischof[2] and Kay Hameyer[1]
1 - Institute of Electrical Machines – RWTH Aachen University
2 - RWTH Aachen University, Center for Computing and Communication
E-mail: Stefan.Boehmer@IEM.RWTH-Aachen.de

***Abstract*—The relative motion between stator and rotor in electrical machines requires a flexible representation in 2D and 3D Finite Element (FE) models. Numerous approaches to incorporate the relative motion are available for the FE method. Along with increasing problem size and accuracy, parallel computing becomes more feasible. The parallel simulation of sufficiently large problems often involves domain decomposition algorithms, especially if distributed memory systems are used for the parallelization. Accounting for motion usually requires explicit domain decomposition at each simulation step. This paper proposes an alternative approach avoiding the computationally expensive domain decomposition at each step by properly choosing an initial decomposition, which can be applied to all steps throughout the simulation.**

I. Introduction

NUMERICAL simulation of electrical machines requires a flexible implementation of the rotor motion in 2D and 3D models. This is especially important for parallelized codes as applied in this paper. Due to the development on the microprocessor market, increase of computational power is not driven by increase of processing frequency anymore but mainly by parallelization [1]. Recent microprocessor architectures are equipped with multiple cores on every processor and additionally with multiple processors per system. Furthermore it is possible to use computing clusters as distributed memory systems for parallelization. Both multiple processor architectures and distributed memory systems require explicit domain decomposition. In this paper the parallelization is implemented in the institute's in-house FE-package iMOOSE [www.iem.rwth-aachen.de] and a hybrid parallelization paradigm based on OpenMP and MPI is deployed. An alternative parallelization particularly designed for vector processors is presented in [2], as well relying on domain decomposition. Several approaches to simulate the motion of electric machines within the FE analysis have been developed. Among them is the moving band technique [3], where an annulus shaped band of elements is re-generated after each rotor movement. Since this approach is only feasible for motion problems in 2D, the lock-step approach is often applied in 3D [4]. The lock-step algorithm replaces the degrees of freedom (DoF) on e.g. the rotor side by the DoFs of the stator side. Therefore, the discretization on the sliding interface must match for each step requiring a fixed motion step size. Lagrange-multiplier approaches seek to overcome the disadvantages being applicable to 2D and 3D, static and transient problems [5].

Almost all approaches accounting for motion have in common, that DoFs on one side are expressed as a linear combination of DoFs on the opposite side. The modified mesh decomposition presented in this paper is compatible with all motion algorithms within this category.

II. Adjusting Parallelization for motion

The parallelization implemented within the iMOOSE library is based on domain decomposition of the FE mesh, so that every involved process is exclusively working on a particular sub mesh. The decomposition of the complete domain Ω into s sub domains Ω_i is given by:

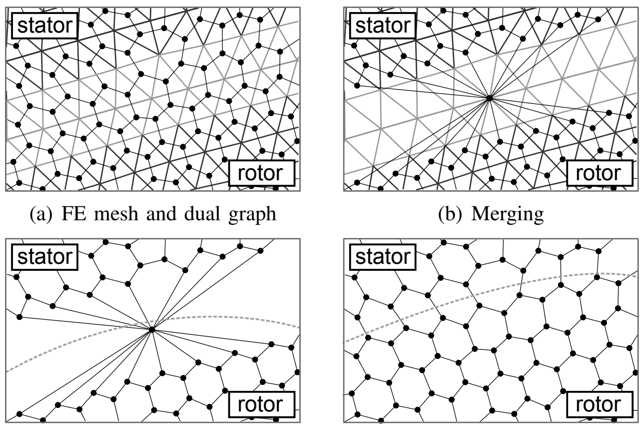
$$\bigcup_{i=0}^{s} \Omega_i = \Omega \quad \text{with} \quad \Omega_i \cap \Omega_j = \emptyset \;\; \text{for} \;\; i \neq j. \tag{1}$$

The decomposition is done by a graph partitioning algorithm, e.g. by multilevel methods [6]. Therefore the dual Graph $G = (V, E, W_v)$ corresponding to the mesh is constructed. V describes the vertex set and $E \subseteq V$ x V the edges. W_v holds the vertex weights. Every vertex in the dual graph represents an element of the mesh. Two vertices are adjacent if and only if the corresponding elements share a common edge in 2D and a common face in 3D. Additionally every vertex is weighted according to the number of DoFs, which is considered by the decomposition algorithm.

If motion is considered for computation, the mesh topology changes every time step at the interface between stator and rotor. The initial mesh decomposition is only valid for the first time step and can not be used for the other ones without modifications. One possible solution is to adjust the decomposition at every step, which results in additional communication by data traffic among all participating processes. To avoid this overhead and minimize process interdependency the mesh decomposition is modified to fit any position of the motion.

III. Modified Domain Decomposition

The presented modified domain decomposition for handling the motion during the parallel execution is based on the idea, that all elements corresponding to the moving interface are assigned to the same sub mesh. To do so, a modification to



(a) FE mesh and dual graph (b) Merging (c) Partitioning (d) Expansion

Fig. 1. Mesh decomposition taking motion into account during the partitioning of the dual graph

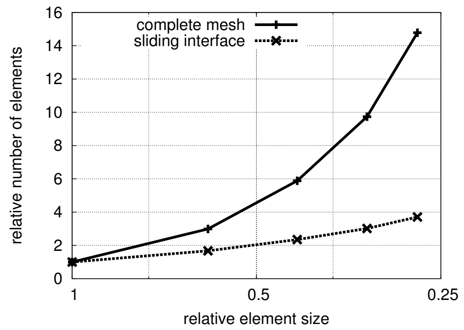


Fig. 2. Number of elements dependent on the mesh size

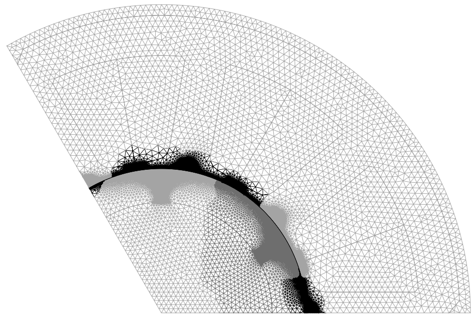


Fig. 3. Decomposition of a 2D PMSM FE mesh into four sub meshes

the standard domain decomposition is required restricting the locality of the motion interface to one sub mesh and thus, significantly reducing the communication.

Let Ω^m and Ω^s be the stator and rotor domain of an electric machine and $\mathcal{T}^m$ and $\mathcal{T}^s$ their triangulations. Let Γ^m and Γ^s be the interface between the stator and rotor domain and $\mathcal{T}^{\Gamma m}$ and $\mathcal{T}^{\Gamma s}$ be the elements belonging to this interface. Initially, a dual graph of the discretization is constructed. Fig. 1(a) shows a zoomed detail of the discretized air gap of a 2D permanent magnet excited synchronous machine (PMSM) FE-model and the corresponding dual graph. In the next step all vertices corresponding to elements from $\mathcal{T}^{\Gamma m} \cup \mathcal{T}^{\Gamma s}$ are merged to one supervertex v^m(Fig. 1(b)). Let V^I contain the vertices corresponding to these elements. The weight of v^m has to be set according to the sum of the weights of the merged vertices in order to avoid an unbalanced partitioning. Thus, a modified graph $G' = (V', E', W'_v)$ is constructed:

$$V' = V \backslash V^I \cup \{v^m\}, \tag{2}$$

$$E' = \{\{v_j, v_k\} \mid \{v_j, v_k\} \in E \wedge v_j \notin V^I \wedge v_k \notin V^I\} \tag{3}$$

$$\cup \{\{v^m, v_j\} \mid \{v_i, v_j\} \in E \wedge v_i \in V^I\}, \tag{4}$$

$$W'_v = \{w_i \in W_v \mid v_i \in V'\} \cup \{w^m\}, \tag{5}$$

$$w^m = \sum_{i, v_i \in V^I} w_i. \tag{6}$$

The graph G' is partitioned by e.g [6]. The resulting cut for the given example graph is shown in Fig. 1(c). Afterwards all vertices in V^I, which have been merged into the supervertex v^m, are expanded by assigning the vertices to the same partition in which the supervertex is located (Fig. 1(d)). Finally, the different sub meshes are determined from the partitioning. Thereby all elements which are located at the moving interface Γ^m and Γ^s are assigned to one single sub mesh.

Only one process has to handle the motion between stator and rotor, thus no additional communication is required for handling the motion. The maximum reasonable number of sub meshes is given by:

$$n_{limit} \leq \frac{|\mathcal{T}^m \cup \mathcal{T}^s|}{|\mathcal{T}^{\Gamma m} \cup \mathcal{T}^{\Gamma s}|}. \tag{7}$$

The limit n_{limit} is reached when a single sub mesh contains all elements of the set $\mathcal{T}^{\Gamma m} \cup \mathcal{T}^{\Gamma s}$. Exceeding this limit of sub meshes leads to unbalanced decompositions being unfavorable and limiting the efficiency of the parallelization. This limit increases with increasing problem size. Fig. 2 shows the number of elements of a complete mesh $|\mathcal{T}^m \cup \mathcal{T}^s|$ compared to the sliding interface $|\mathcal{T}^{\Gamma m} \cup \mathcal{T}^{\Gamma s}|$ in function of different element sizes for a 2D field problem. It can be observed, that the slope of the relative overall number of elements is quadratically with respect to the relative element size, while the slope of the number of elements on the sliding interface increases almost linearly, since this interface is a one dimensional sub mesh. Thus, the theoretical limit n_{limit} is unlikely to be reached when simulating electrical machines using the current approach.

IV. Application

The described method has been applied to a 2D quasi-static field problem of a PMSM. Fig. 3 shows the result of the mesh decomposition into four different sub meshes for the parallel computation with four processes. The black colored sub mesh contains all the elements of the set $\mathcal{T}^{\Gamma m} \cup \mathcal{T}^{\Gamma s}$ as a subset. For this mesh, which contains 17670 DoFs, the maximum reasonable number of sub meshes computed by (7) yields 21.

V. Conclusion

This paper proposes a modification to the standard domain decomposition in parallel computing of electrical machines. By introducing a weighted, virtual supervertex in the dual graph of the elements on the sliding air gap interface, single domain decomposition yields proper sub meshes for all subsequent simulation steps. Hereby, an efficient parallel computation of electrical machines accounting for motion on distributed memory system as well as on multi processor architectures can be implemented.

The full paper will contain an application of the proposed approach to a 3D field problem and measurements of the speedup ratio, which is achieved by this approach.

References

[1] M. J. Flynn and P. Hung, "Microprocessor design issues: Thoughts on the road ahead," *IEEE Micro*, vol. 25, pp. 16–31, May 2005.

[2] T. Nakano, Y. Kawase, T. Yamaguchi, M. Nakamura, N. Nishikawa, and H. Uehara, "Parallel computing of magnetic field for rotating machines on the earth simulator," *Magnetics, IEEE Transactions on*, vol. 46, no. 8, pp. 3273–3276, 2010.

[3] B. Davat, Z. Ren, and M. Lajoie-Mazenc, "The movement in field modeling," *Magnetics, IEEE Transactions on*, vol. 21, no. 6, pp. 2296–2298, 1985.

[4] T. Preston, A. Reece, and P. Sangha, "Induction motor analysis by time-stepping techniques," *Magnetics, IEEE Transactions on*, vol. 24, no. 1, pp. 471–474, 1988.

[5] E. Lange, F. Henrotte, and K. Hameyer, "A variational formulation for nonconforming sliding interfaces in finite element analysis of electric machines," *Magnetics, IEEE Transactions on*, vol. 46, no. 8, pp. 2755–2758, 2010.

[6] G. Karypis and V. Kumar, "Multilevel algorithms for multi-constraint graph partitioning," in *Proceedings of the 1998 ACM/IEEE conference on Supercomputing*, ser. Supercomputing '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 1–13.