Multiply Connected 3D Transient Problem with Rigid Motion Associated with $\boldsymbol{T}$- $\boldsymbol{\Omega}$ Formulation

C. Lu, P. Zhou, D. Lin, B. He and D. Sun
Ansys Inc, 225 W. Station Square Dr., Pittsburgh 15219 USA Chuan.Lu@ansys.com

***Abstract* — An automatic cut generation algorithm based on the tree/co-tree technique is applied to treat multiply connected regions in 3D transient solutions including rigid motion. In the cutting domains, the zero curl condition of vector potential $\boldsymbol{T}$ is strongly imposed. A new algorithm is developed to guarantee that the generation of every cutting domain will reside on either the stationary region or on the moving region without touching or crossing the mesh coupling interface.**

***Index Terms*—Finite element methods, Time domain analysis, Tree graphs**

I. Introduction

The $\boldsymbol{T}$ - $\boldsymbol{\Omega}$ formulation is a very efficient formulation for solving eddy-current problems, especially for transient simulation. Here the scalar potential $\boldsymbol{\Omega}$ is normally represented by nodal shape functions in the entire domain while the current vector potential $\boldsymbol{T}$ represented by edge-based shape functions is restricted to the conducting regions only. When the conducting regions contain holes, the non-conducting region becomes multiply connected and the scalar potential may become multi-valued. The topic of treating multiply connected regions has drawn considerable attention [1]-[4]. An efficient one is to apply the tree-cotree technique to make multiply connected regions simply connected for both the source field computation [3] and eddy current computations [2],[4]. To further take rigid motion into account during a transient simulation, a separation technique is introduced to confine the generation of each cutting domain to either the stationary or moving regions for non-conforming mesh coupling and reducing computational cost [5]. However, since the cutting domain generation is an automatic and random process, and the co-tree may spread across the mesh coupling interface, which leads to the failure of cutting domain generation.

In this paper, a new algorithm is developed to guarantee the generation of every cutting domain will reside on either the stationary region or on the moving region without touching or crossing the mesh coupling interface. In addition, a very convenient and reliable scheme is introduced to handle periodic boundary conditions.

II. Concept of Automatic Cutting Domain Creation

For the completeness, let us first have a brief review of some basic concept about automatic generation of cutting domain. It is far from obvious to manually locate the cuts even for very experienced designers. The automatic generation of cuts is based on the automatic identification of tree edges and co-tree edges for every element. If an edge has been defined either as a branch of the tree or as a chord of the co-tree, the edge is considered as "identified". If an edge is the third side of a triangle whose other two sides has been identified, this third edge will be identified as a chord and close the three-edge loop of the triangle. Otherwise, there are two cases to consider. If there is a path of previously-identified edges connecting one end of the edge to the other, this edge cannot be identified since this would form a closed path which might violate Ampere's law. On the other hand, if there is no such connecting path, the edge can be safely identified. If all three edges of a triangle have been identified, the triangle becomes single connected [5].

The above rules are applied to the automatic generation of cutting domains. In a bounded region R that is the union of conducting region $\boldsymbol{R}_{\mathrm{c}}$ and non-conducting region $\boldsymbol{R}_{\mathrm{n}}$, we search for a maximum set of tetrahedrons $\boldsymbol{R}_{\mathrm{m}}$ in $\boldsymbol{R}_{\mathrm{n}}$ such that the set $\boldsymbol{R}_{\mathrm{m}}$ is simply connected instead of detecting the holes and looking directly for the cutting domains. By saying maximum, we mean that the domain $\boldsymbol{R}_{\mathrm{m}}$ becomes multiply connected by adding any additional tetrahedron to the set $\boldsymbol{R}_{\mathrm{m}}$. Once such a set is determined, the remaining tetrahedrons in $\boldsymbol{R}_{\mathrm{n}}$ form the cutting domain $\boldsymbol{R}_{\Sigma}$ (=$\boldsymbol{R}_{\mathrm{n}}\backslash$ $\boldsymbol{R}_{\mathrm{m}}$). Such a cutting domain is not unique but works for our purpose. The procedure for generating cutting domains consists of two steps: first, identify as much edges as possible by the above rules on the surface of conducting regions. Second, continue to identify as much edges as possible in the non-conducting region. If all six edges of a tetrahedron are identified, we add this tetrahedron to the set of singly connected domain. Finally, the remaining tetrahedrons that are not included in the singly connected domain create the desired cutting domains $\boldsymbol{R}_{\Sigma}$.

III. A New Algorithm to Avoid Cut across the Mesh Coupling Interface

When motion is involved, two independent meshes must be coupled together after an arbitrary displacement of the moving parts. The coupling of the meshes between the moving parts and the stationary parts is handled by either the sliding surface method or the moving band method depending on the motion type. For the sliding surface method, the stationary and moving meshes are coupled together along the sliding interface between them. For the moving band method, an additional band region is used to separate the stationary and moving parts and only the mesh in the band region is recreated at each time step. To achieve maximum flexibility and good discretization quality, non-conforming meshes are used for the coupling in both cases. This means that the scalar potential $\boldsymbol{\Omega}$ at each node, the source field component $\boldsymbol{H}_p$ and vector potential $\boldsymbol{T}$ at each edge on the slave surface have to be mapped onto the master surface to eliminate all unknowns on the slave surface. Finding a general algorithm for mapping node-based scalar potential unknowns should not be too difficult. In the case of mapping vector unknowns, however, the process of splitting slave edge variables with respect to the trace of the master mesh while preserving the valid cutting domains is not impossible but very complicated.

To overcome this difficulty, a separation technique is applied to confine the generation of every cutting domain to either the stationary region or the moving region without crossing the mesh coupling interface [5]. As a result, the process of splitting slave edges with respect to the trace of the master mesh for mapping edge-based vector potentials is completely eliminated and only the node-based scalar potential is involved. It follows that we need to develop an algorithm to ensure the generation of every cutting domain will reside on either the stationary region or on the moving region even though this generation process is an automatic and random process.

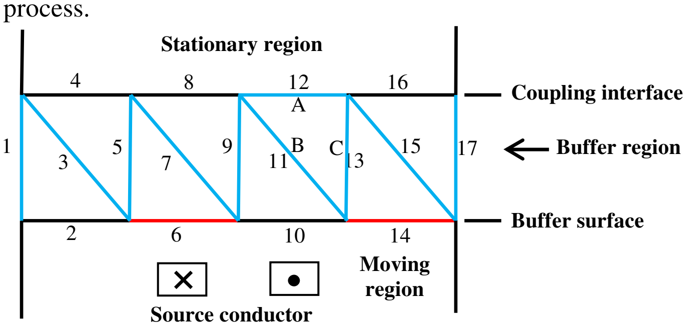

Fig. 1. Schematic representation of buffer region and $\boldsymbol{H_p}$ assignment

For easy explanation, let us take $\boldsymbol{H_p}$ assignment as example and first define a buffer region which consists in one layer of elements neighboring the mesh coupling interface towards the source conductors under discussion, as shown in Fig. 1. Here we also name the other surface of the buffer region as "buffer surface" that is closer to source conductors than the mesh "coupling interface". Our goal here is to introduce an algorithm to make all the nonzero $\boldsymbol{H_p}$ (cutting edge) assignment to be completed through the buffer region at most without propagating further to touch the coupling surface. This algorithm can be described in terms of the following steps as:

1. First make $\boldsymbol{H_p}$ assignment on conductor surface, which includes the nonzero $\boldsymbol{H_p}$ identification of a band of cutting edges for each conductor. Then extend the surface identification to the non-conducting region enclosed by the buffer surface. At this point, the $\boldsymbol{H_p}$ identification has completed inside the region enclosed by the buffer surface and it normally happens that on the buffer surface, there are some traces of cutting edges with nonzero $\boldsymbol{H_p}$ value, as edge 6 and 14 in Fig. 1 in a 2D view.
2. Do the $\boldsymbol{H_p}$ identification solely on the mesh coupling surface. The outcome is that the identification on all edges should have been completed except leaving one band of edges (cutting edges) unidentified when taking into account the periodical boundary condition. This one band of unidentified edges degrades to one edge in a 2D view, as edge 12 in Fig 1.
3. Create a restore point for possible recovery later by storing all the index of nodes, edges, triangles and tetrahedrons.
4. Identify all the edges associated with the buffer region. For the example in Fig. 1, to be identified edges include edges 1, 3, 5, 7, 9, 11, 13 and 15 (edge 17 is the same edge as edge 1 due to periodical condition) inside the buffer region and edge A (12) on the mesh coupling interface. Start the identification process from an arbitrary internal edge with zero $\boldsymbol{H_p}$ and complete the identification for all the above edges. As a result, there are two possibilities: one is if edge A is identified as an edge with zero $\boldsymbol{H_p}$ value, this cutting domain is successfully created and the process stops since all the nonzero $\boldsymbol{H_p}$ cutting edges has been confined to inside the buffer region without propagating further to across the mesh coupling surface; otherwise, edge A has been identified as a cutting edge with nonzero $\boldsymbol{H_p}$ value. In such a case, an additional step 5 is required:
5. Recover all the index of nodes, edges, triangles and tetrahedrons stored in step 3 and choose a new internal edge (edge B or edge C) as starting edge with the nonzero $\boldsymbol{H_p}$ value on edge *A* derived in step 4. In this way, it is guaranteed that edge A will be identified as a non-cutting edge with zero $\boldsymbol{H_p}$ value and all cutting edges with nonzero $\boldsymbol{H_p}$ will be successfully confined to inside the buffer region without propagating further to across the mesh coupling surface.

IV. Application Example

The example is a 538 KW three phase synchronous motor. This motor has 6 poles, 72 stator slots. Its Y-connected 3-phase winding is energized by 400 line-to-line volts at 50 Hz. For the cage, 7 cuts associated with 7 holes are identified automatically even if one hole is cut into two halves. Figure 2 shows **B** plot in stator and rotor, the induced eddy currents in bars and source currents in stator windings at t = 0.0464s.

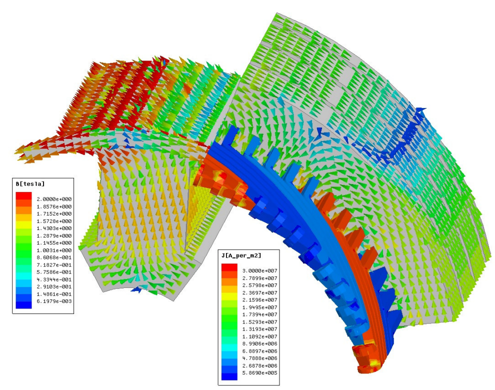

Fig. 2. Plot of flux density, induced eddy currents and source currents

References

[1] Leonard P.J., Hill-Cottingham and Roger D., "3D finte element models and external circuits using the AΨ scheme with cuts", *IEEE Trans. Magnetics*, 30(5):3220-3223, 1994.

[2] Henrotte F. and Hameyer K., "An algorithm to construct the discrete cohomology basis functions required for magnetic scalar potential formulations without cuts", *IEEE Trans. Magnetics*, 39(3):1167-1170, 2003.

[3] Ren Z., Zhou P. and Cendes Z.J., "Computation of current vector potential due to excitations in multiply connected conductors," in *Proc. ICEF Int. Conf.*, Tianjin, China, 2000, pp. 121-124.

[4] Ren Z., "***T***-Ω formulation for eddy-current problems in multiply connected regions", *IEEE Trans. Magnetics*, 38(2):557-560, 2002.

[5] P. Zhou, Z. Badics, D. Lin, and Z. J. Cendes, "Nonlinear ***T***- **Ω** Formulation Including Motion for Multiply Connected 3-D Problems," *IEEE Trans. Magn.* Vol. 44, no. 6, pp. 718-721, Jun. 2008