Progress In Electromagnetics Research, Vol. 160, 1–8, 2017

Time Decomposition Method for the General Transient Simulation of Low-Frequency Electromagnetics

Bo He*, Chuan Lu, Ningning Chen, Dingsheng Lin, Marius Rosu, and Ping Zhou

Abstract—This paper describes a highly robust and efficient parallel computing method for the transient simulation of low-frequency electromagnetics with nonlinear materials and/or permanent magnets. In this method, time subdivisions are introduced to control the memory usage and nonlinear convergence. A direct block triangular matrix solver is applied to solve the formulated block matrix for each subdivision. This method has been implemented using the Message Passing Interface (MPI) for distributed memory parallel processing. Depending on the number of available MPI processes and physical memory, the entire nonlinear transient simulation can be divided into several subdivisions along the time-axis such that each MPI process handles only the computation for one time-step. Application examples are presented to demonstrate that this method can achieve excellent scalability of speedup.

1. INTRODUCTION

Electric machines, power and electronic transformers can be better designed and analyzed using transient electromagnetic field simulation. This choice allows engineers to analyze the dynamic system including the nonlinear materials, permanent magnets and induced eddy currents under a variety of conditions, under various excitations including the pulsed waveform. This procedure is usually very time-consuming since it requires $N_t \cdot N_e$ matrix solutions, where N_t is the number of time steps and N_e is the average number of nonlinear iterations [1–6]. Provided that an algorithm (or method) can be made parallel, parallel computing can cut down simulation time for a nonlinear transient problem. For example, parallel computing can be applied to the matrix assembling and matrix solving at each time step [7]. However, it is not always possible to make full use of all the parallel cores because of limited parallel scalability. In order to gain better parallel scalability, an approach based on an iterative solver with incomplete LU factorizations is proposed in [8], but it is not very robust for real engineering applications because the iterative solver may fail to converge for a given accuracy. Also, incomplete LU factorizations have the limitations of potential instabilities and lack of algorithmic scalability [9]. Based on a block direct solver, we proposed a highly robust and scalable parallel computing method, called the time decomposition method (TDM) for general transient simulation of low-frequency electromagnetics [10]. In this paper, we will discuss this scheme extensively and demonstrate its effectiveness by presenting several application examples.

2. TIME DECOMPOSITION METHOD

2.1. Fundamental Equations

In low-frequency applications, the finite element method discretization of Maxwell equations with nonlinear materials produces the following semi-discrete form of equations:

$$S(x,t)\,x(t) + \frac{d}{dt}\left[T(x,t)\,x(t)\right] = f(x,t) + \frac{d}{dt}\left[w(x,t)\right] \tag{1}$$

Received 25 July 2017, Accepted 7 October 2017, Scheduled 22 October 2017
* Corresponding author: Bo He (bo.he@ansys.com).
The authors are with the Ansys Inc., 2600 Ansys Drive, Canonsburg, PA 15317, USA.

In low-frequency electromagnetics, two typical formulations are often used, the A-φ and T-Ω formulations, respectively [11]. In the A-φ formulation, $T\left(x,t\right)$ is a constant, so Eq. (1) can be simplified to

$$S\left(x,t\right)x\left(t\right)+T\frac{d}{dt}x\left(t\right)=f\left(x,t\right)+\frac{d}{dt}\left[w\left(x,t\right)\right] \tag{2}$$

Equation (2) is similar to the formulation presented in [8]. However, in the T-Ω formulations, $T\left(x,t\right)$ is dependent on the solution vector $x\left(t\right)$ because of the non-linearity, so Eq. (1) must be used. In this paper, all simulations have been performed based on T-Ω formulations. Applying the backward Euler method and the Newton-Raphson method to Eq. (1), we have the following linearized matrix equations, written in the form of a block matrix

$$\begin{bmatrix} K_1 & 0 & \cdots & 0 & 0 \\ M_1 & K_2 & \cdots & 0 & 0 \\ 0 & M_2 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & K_{n-1} & 0 \\ 0 & 0 & \cdots & M_{n-1} & K_n \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_{n-1} \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} \tag{3}$$

with each submatrix corresponding to a single time-step. In the above, $K_i=\Delta t S_i'+T_i'$ and $M_i=-T_i'$ are the Jacobian matrices; Δx_i is the increment of the solution during nonlinear iterations; b_i is the residual during nonlinear iterations. =

2.2. Subdivisions

Theoretically, either a general purpose direct solver [12] or an iterative solver [13, 14] can be applied to solve Eq. (3). However, for real engineering problems, this might be either very time-consuming and/or memory prohibitive. In order to tackle this issue, the entire nonlinear transient simulation is divided into several subdivisions along the time-axis such that a smaller block matrix needs to be solved at a time. One example is shown in Fig. 1. Subdivision 1 handles the computation for time step 1 to m and subdivision 2 handles the computation for time step $m+1$ to $2m$. After solving one subdivision, the solution of the final time step of current subdivision is used as the initial conditions to solve the next subdivision.

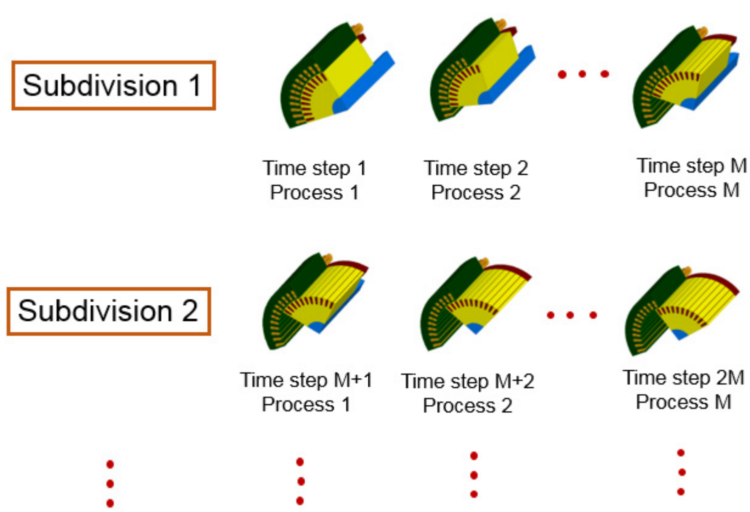

Figure 1. Schematic of subdivisions (the solution of the final time step of subdivision 1 is used as the initial conditions to solve the subdivision 2).

2.3. Direct Block Triangular Matrix Solver

For each subdivision, block matrixes, i.e.,

$$\begin{bmatrix} K_1 & 0 & \cdots & 0 & 0 \\ M_1 & K_2 & \cdots & 0 & 0 \\ 0 & M_2 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & K_{m-1} & 0 \\ 0 & 0 & \cdots & M_{m-1} & K_m \end{bmatrix} \tag{4}$$

must be assembled and solved at each Newton-Raphson iteration. Note that m is usually either smaller or much smaller than n. Since Eq. (4) is a lower block triangular matrix, the most efficient and robust solver is a direct block triangle matrix solver. The basic procedure of the direct block triangle matrix solver is presented in Fig. 2. It includes two parts, submatrix LU factorization and block forward substitutions. The first part, submatrix LU factorization, can be done independently in parallel while the second part, block forward substitutions, is essentially sequential. In terms of computational time, the first part is dominant over the second part, so this direct block triangle matrix solver naturally facilitates parallel computing.

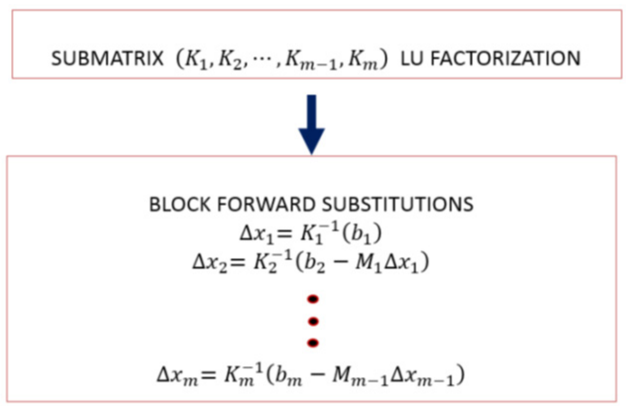

Figure 2. Direct block triangular matrix solver.

2.4. Distributed Memory Parallel

There are two fundamental approaches to parallel computing: distributed memory parallel and shared memory parallel. This proposed method can be naturally implemented as distributed memory parallel using the Message Passing Interface (MPI) [15]. Another advantage of the distributed memory parallel is that it is not limited to one computer. Theoretically, for a given hardware resource, there are several ways to do subdivisions. The most efficient way is that the available cores should be sufficiently used and in the meantime, each MPI process handles only the computation of one time-step. In this way, the nonlinear system to be solved is minimized such that better memory usage is achieved. It also gives a better nonlinear convergence since the larger the nonlinear system the more difficult it is to converge. The FEM matrix assembling and the post processing for each subdivision can also be done in distributed memory parallel framework. The basic scheme of the distributed memory parallel approach at each subdivision is presented in Fig. 3.

3. APPLICATION EXAMPLES

The proposed TDM has been applied to the transient simulations of different types of electrical machines, transformers and other magnetic devices. All the computations of the following three application examples were performed on a computer cluster with 16 nodes, in which each node consists of 28 cores and 512 GB RAM.

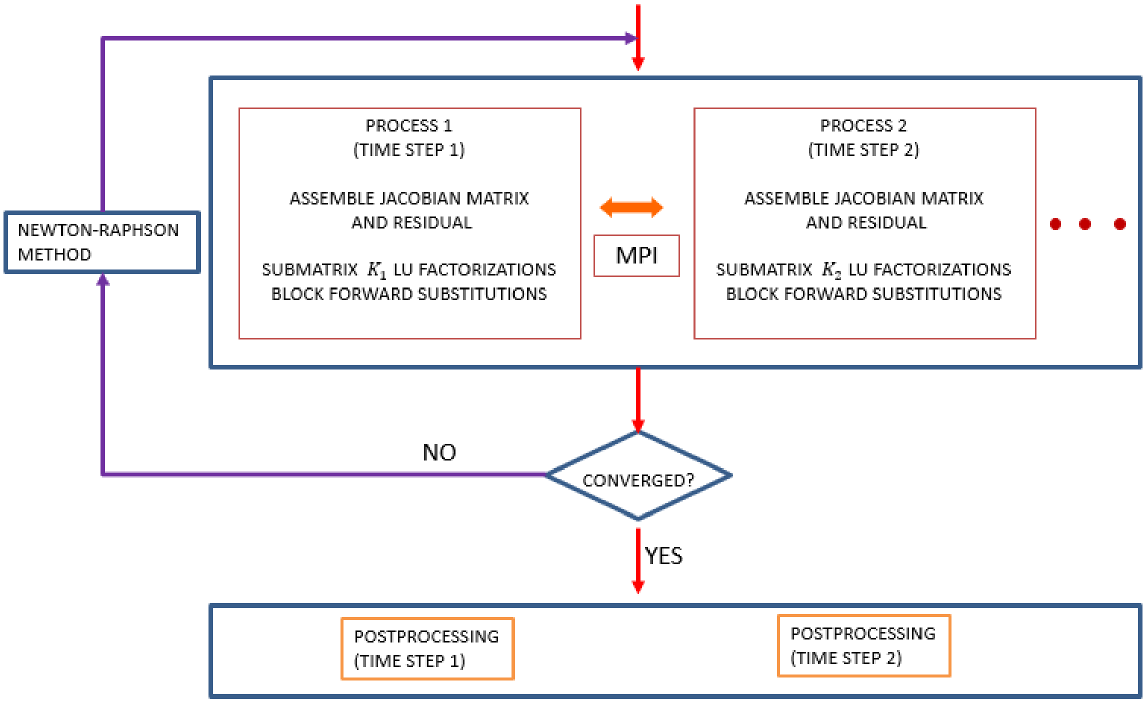

Figure 3. The distributed memory parallel approach based on MPI for one subdivision.

3.1. Permanent Magnet Motor

Figure 4 shows a three-phase permanent motor with nonlinear-magnetic materials for both rotor and stator. The speed is 1800 RPM and driven frequency is 60 Hz. The induced eddy-current effect is considered in magnet. Fig. 5 shows the input three-phase voltages. The number of mesh elements is 124148. The total number of time steps is 256. Table 1 presents the parallel computation performance of this permanent magnet motor. The speedup is calculated against the sequential case without distributed parallel computing denoted as one MPI process. For 2 MPI processes, there is no speedup for this case. This is because for the sequential case, the solution at the previous time step can be used to estimate the initial nonlinear operating point for the first nonlinear iteration of the next time step. This speeds up the convergence for the sequential case but for the TDM parallel case this scheme is no longer applicable. However, with the increase of the number of MPI processes, the parallel speedup is obvious. The speedup for 256 MPI processes is 13.0.

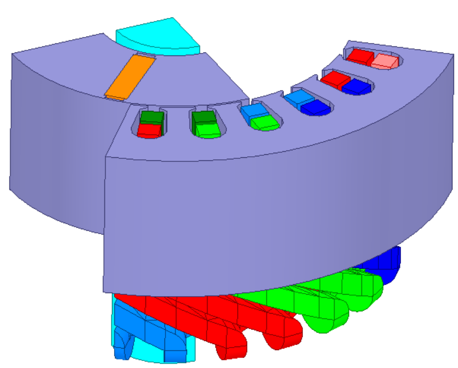

Figure 4. Permanent magnet motor.

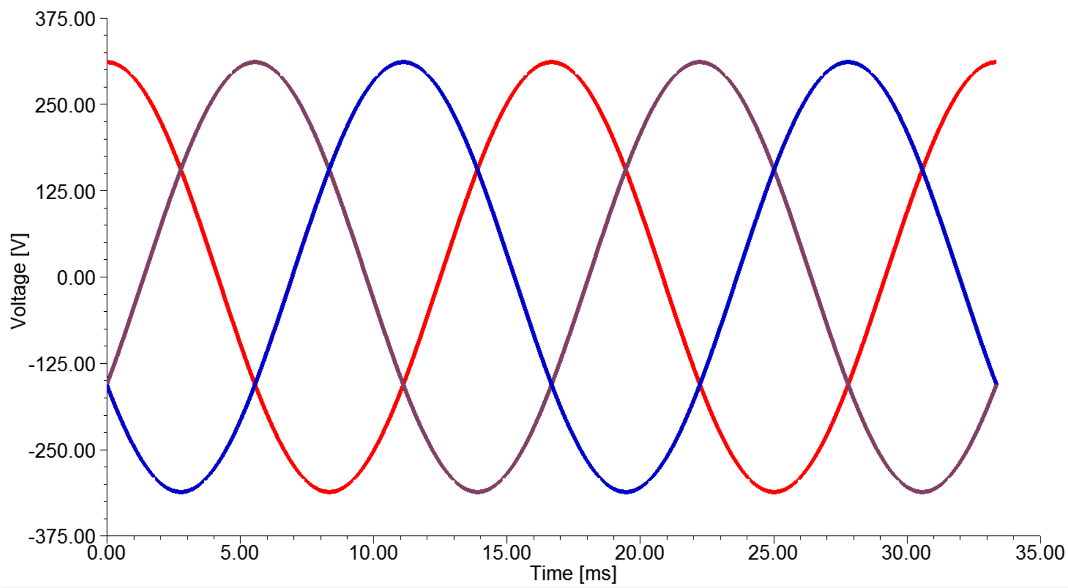

Figure 5. The input three-phase voltage sources.

Table 1. Performance of TDM for permanent magnet motor.

Number of MPI processes	Number of subdivisions	Simulation time (hours:minutes:seconds)	Speedup
1	256	(9:19:45)	1
2	128	(11:51:31)	0.79
4	64	(06:32:48)	1.43
8	32	(04:28:46)	2.08
16	16	(02:42:21)	3.45
32	8	(01:57:3)	4.78
64	4	(01:18:35)	7.12
128	2	(00:59:11)	9.46
256	1	(00:43:12)	13.0

3.2. Double Cage Induction Motor

Figure 6 shows an 8-pole double cage induction motor. The slip is 0.0334. The speed is 724.979 RPM and driven frequency is 50 Hz. The induced eddy-current effect is considered in the copper damping bar. Fig. 7 shows the input three-phases voltages. The total number of time steps is 256. The number of mesh elements is 480942. Table 2 presents the parallel efficiency. In Table 2, the speedup for 256 MPI processes is 26.7.

3.3. Planar Transformer

Figure 8 shows a planar transformer with a nonlinear magnetic core. Fig. 9 shows the input voltage with the pulsed waveform. The induced eddy-current effect is considered in the multi-turn planar winding. The traditional time stepping method is very time-consuming due to large time constants and large matrix size. The total number of time steps is 256. The number of mesh elements is 875927. Table 3 gives the parallel efficiency. The speedup for 256 MPI processes is 15.4.

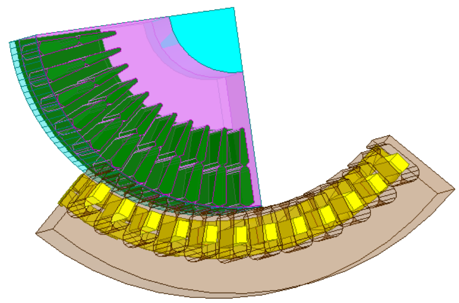

Figure 6. Double cage induction motor.

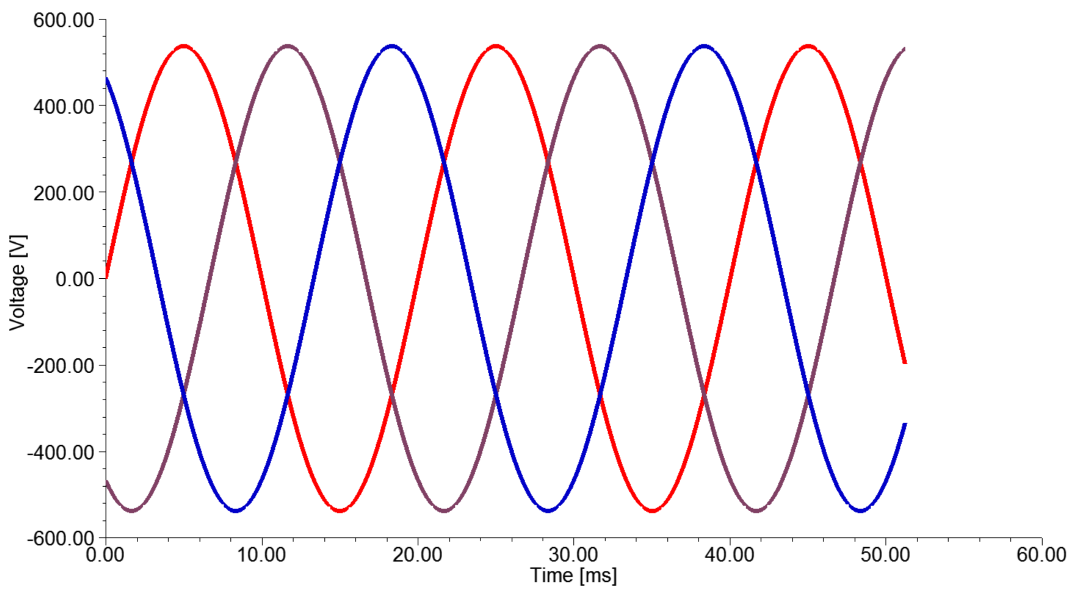

Figure 7. The input three-phase voltage sources.

Table 2. Performance of TDM for double cage induction motor.

Number of MPI processes	Number of subdivisions	Simulation time (hours:minutes:seconds)	Speedup
1	256	(104:36:21)	1
2	128	(166:28:46)	0.63
4	64	(90:52:05)	1.15
8	32	(47:36:23)	2.20
16	16	(29:37:52)	3.53
32	8	(16:17:27)	6.42
64	4	(9:08:58)	11.4
128	2	(6:22:14)	16.4
256	1	(3:55:25)	26.7

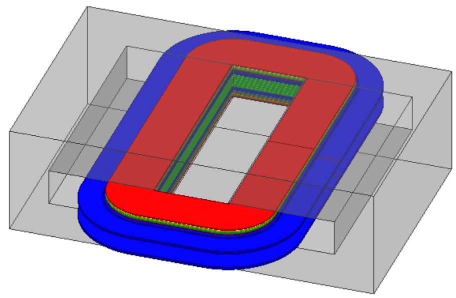

Figure 8. Planar transformer.

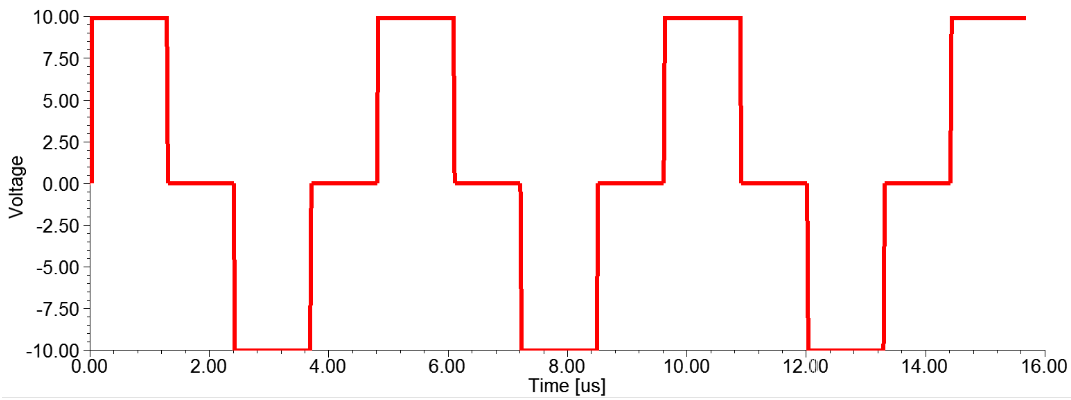

Figure 9. The input voltage with pulsed waveform.

Table 3. Performance of TDM for planar transformer.

Number of MPI processes	Number of subdivisions	Simulation time (hours:minutes:seconds)	Speedup
1	256	(34:28:28)	1
2	128	(25:04:44)	1.37
4	64	(13:08:30)	2.62
8	32	(7:30:51)	4.59
16	16	(5:34:05)	6.19
32	8	(3:20:55)	10.3
64	4	(3:43:50)	9.24
128	2	(3:25:28)	10.1
256	1	(2:14:31)	15.4

4. CONCLUSIONS

A highly robust and efficient parallel computing method for the transient simulation of low-frequency electromagnetics is proposed. In this method, time subdivisions are introduced to control the memory usage and nonlinear convergence, and a direct block triangular matrix solver is applied to solve the formulated block matrix for each subdivision. Application examples are presented to demonstrate that this method can achieve more than 10 times speedup.

REFERENCES

1. Yan, S., J.-M. Jin, C.-F. Wang, and J. D. Kotulski, "Numerical study of a time-domain finite element method for nonlinear magnetic problems in three dimensions (invited paper)," *Progress In Electromagnetics Research*, Vol. 153, 69–91, 2015.
2. Yan, S. and J.-M. Jin, "Theoretical formulation of a time-domain finite element method for nonlinear magnetic problems in three dimensions (invited paper)," *Progress In Electromagnetics Research*, Vol. 153, 33–55, 2015.
3. Pries, J. and H. Hofmann, "Steady-state algorithms for nonlinear time-periodic magnetic diffusion problems using diagonally implicit Runge-Kutta methods," *IEEE Trans. Magn.*, Vol. 51, No. 4, 1–12, Apr. 2015.
4. Zhou, P., W. N. Fu, D. Lin, S. Stanton, and Z. J. Cendes, "Numerical modeling of magnetic devices," *IEEE Transactions on Magnetics*, Vol. 40, No. 4, 1803, 2004.
5. Demerdash, N. A., J. F. Bangura, and A. A. Arkadan, "A time-stepping coupled finite element-state space model for induction motor drives — Part 1: Model formulation and machine parameter computation," *IEEE Trans. Energy Conversion*, Vol. 14, 1465–1471, Dec. 1999.
6. Pries, J., "Computationally efficient steady-state simulation algorithms for finite-element models of electric machines," Ph.D. Dissertation, University of Michigan, 2015.
7. Kumbhar, A., K. Chakravarthy, R. Keshavamurthy, and G. V. Rao, "Utilization of parallel solver libraries to solve structural and fluid problems," *Intel Math Kernel Library — White Papers*, 2011.
8. Takahashi, Y., T. Tokumasu, M. Fujita, T. Iwashita, H. Nakashima, S. Wakao, and K. Fujiwara, "Time-domain parallel finite-element method for fast magnetic field analysis of induction motors," *IEEE Trans. Magn.*, Vol. 49, No. 5, 2413–2416, May 2013.
9. Benzi, M., "Preconditioning techniques for large linear systems: A survey," *J. Computat. Phys.*, Vol. 182, 418–477, 2002.
10. He, B., P. Zhou, C. Lu, N. Chen, D. Lin, M. Rosu, and B. Bork, "A time decomposition method for the transient simulation of low-frequency electromagnetics," *Proc. 17th Biennial IEEE CEFC*, Miami, FL, USA, WO23-5, 2016.
11. Kriezis, E. E., T. D. Tsiobukis, S. M. Panas, and J. A. Tegopoulos, "Eddy currents: Theory and applications," *Proc. IEEE*, Vol. 80, 1559–1589, Oct. 1992.
12. Davis, T. A., *Direct Methods for Sparse Linear Systems*, SIAM, Philadelphia, PA, 2006.
13. Van der Vorst, H. A., *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press, 2003.
14. Saad, Y. and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Statist. Comput.*, Vol. 7, No. 3, 856–869, 1986.
15. Gropp, W., E. Lusk, N. Doss, and A. Skjellum, "A high-performance portable implementation of the MPI message-passing interface standard," *Parallel Computing*, Vol. 22, No. 6, 789–828, 1996.