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

B. He, P. Zhou, C. Lu, N. Chen, D. Lin, and M. Rosu

2600 Ansys Drive, Canonsburg, PA 15317 USA, bo.he@ansys.com

This paper describes a highly robust and efficient parallel computing method for the transient simulation of low-frequency electromagnetics. In this method, time subdivision is introduced to control the memory usage and nonlinear convergence and a block forward substitution method is applied to solve the formulated block matrix for each subdivision. Application examples are presented to demonstrate the effectiveness of this method.

***Index Terms*—Transient analysis, Finite element method, Parallel computing.**

I. Introduction

The transient simulation of low-frequency electromagnetics with non-linear materials and permanent magnets usually is time-consuming since it requires $N_t N_e$ number of matrix solutions, where N_t is the number of time steps and N_e is the average number of nonlinear iterations [1][2][3][4]. 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 [5]. 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 is proposed in [6], but it is not very robust for real engineering applications because the iterative solver may fail to converge for a given accuracy. 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 [7]. In this paper, we will discuss this scheme extensively, and demonstrate its effectiveness by presenting several numerical examples.

II. Time Decomposition Method

The finite element method discretization of nonlinear eddy current problems produces a semi-discrete form as

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

Note that $S(x,t)$ and $T(x,t)$ are dependent on solution vector $x(t)$ to reflect the non-linearity of the eddy current problems. Applying the backward Euler method and the Newton-Raphson method, we have the following linearized matrix equations, written in the form of block matrix

$$\begin{bmatrix} K_1 & 0 & \dots & 0 & 0 \\ M_1 & K_2 & \cdots & 0 & 0 \\ 0 & M_2 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & K_{n-1} & 0 \\ 0 & 0 & \cdots & M_{n-1} & K_n \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_{n-1} \\ \Delta x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} \qquad (2)$$

with each submatrix corresponding to each 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 solution during nonlinear iterations, and b_i is the residual during nonlinear iterations. Solving (2) using a general purpose direct matrix solver for real engineering problems is prohibitive. In order to solve (2), we introduced TDM such that distributed parallel computing can be leveraged as described below.

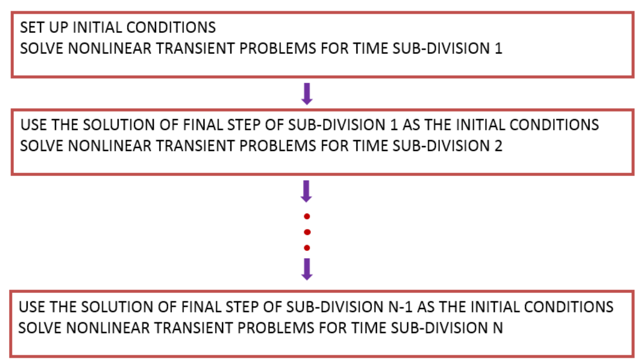

Fig. 1. Flowchart of time decomposition method.

The basic procedure of the time decomposition (TDM) is presented in Fig. 1. It can be implemented based on message passing interface (MPI). Depending on the number of available MPI processes and physical memory, divide the entire nonlinear transient simulation into several sub-divisions along the time-axis such that each MPI process handles only the computation of one time-step. The communication between different time-steps is through MPI functions. For each subdivision, one needs to solve a portion of the block matrix (2), i.e.,

$$\begin{bmatrix} K_1 & 0 & \dots & 0 & 0 \\ M_1 & K_2 & \cdots & 0 & 0 \\ 0 & M_2 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & K_{m-1} & 0 \\ 0 & 0 & \cdots & M_{m-1} & K_m \end{bmatrix} \qquad (3)$$

Since (3) is a lower block triangular matrix, the most efficient solver is a block direct solver as presented in Fig. 2.

After solving one subdivision, use the solution of the final time step of this subdivision as the initial conditions to solve the next subdivision.

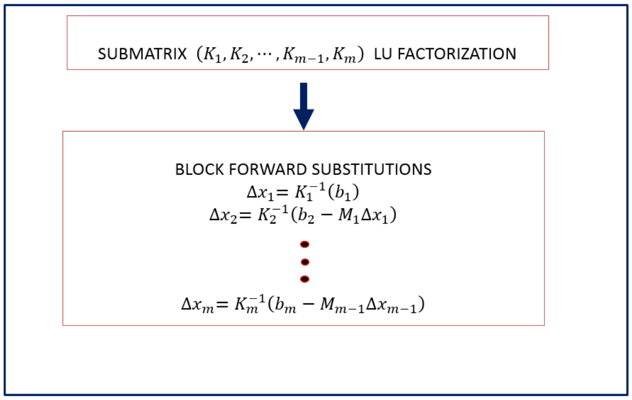

Fig. 2. Direct block triangular matrix solver.

III. Application Examples

The proposed TDM has been applied to the transient simulations of different types of electrical machines, transformers and other magnetic devices.

A. *Permanent magnet motor*

Fig. 3 shows a permanent magnet motor. The speed is 2000 RPM and driven frequency is 200 Hz. The total number of time steps is 256. The number of mesh elements is 887275. Table I gives the parallel efficiency. The speedup is calculated against the sequential case without distributed parallel computing. In table I, the speedup is 21.7 for 256 MPI processes.

B. *Double cage induction motor*

Fig. 4 shows an 8-poles double cage induction motor. The slip is 0.0334. The speed is 724.979 RPM and driven frequency is 50 Hz. The total number of time steps is 256. The number of mesh elements is 480942. Table II presents the parallel efficiency. In table II, the speedup is 26.7 for 256 MPI processes.

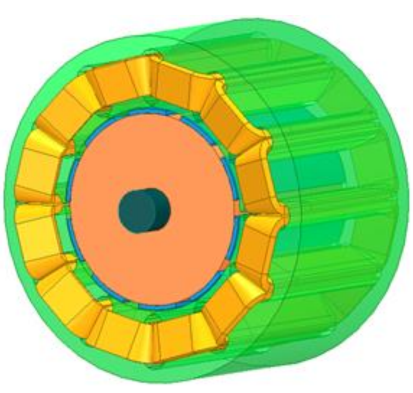

Fig. 3. Permanent Magnet Motor.

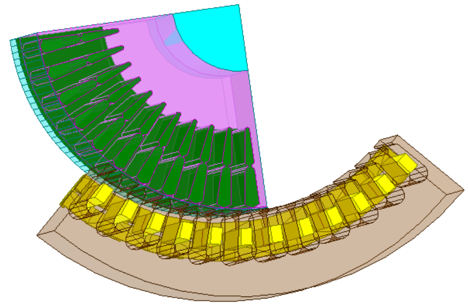

Fig. 4. Double cage induction motor.

TABLE I
Performance of TDM for permanent magnet motor

Number of MPI processes	Number of subdivisions	Simulation time (hours:minutes:seconds)	Speedup
1	256	(49:12:06)	1
2	128	(34:57:25)	1.41
4	64	(19:37:02)	2.5
8	32	(11:37:11)	4.23
16	16	(7:42:48)	6.38
32	8	(4:47:38)	10.3
64	4	(3:26:35)	14.3
128	2	(2:50:49)	17.3
256	1	(2:16:18)	21.7

TABLE II
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

References

[1] J. Pries, 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, pp. 1-12, Apr. 2015.

[2] P. Zhou, W. N. Fu, D. Lin, S. Stanton, and Z. J. Cendes, "Numerical Modeling of Magnetic devices", IEEE Transactions on Magnetics, Volume.40, Issue.4, pp.1803, 2004.

[3] N. A. Demerdash, J. F. Bangura, 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, pp. 1465-1471, Dec. 1999.

[4] J Pries, *Computationally Efficient Steady–State Simulation Algorithms for Finite–Element Models of Electric Machines*, Ph.D. dissertation, University of Michigan., 2015.

[5] A. Kumbhar, K. Chakravarthy, R. Keshavamurthy, G. V. Rao, "Utilization of Parallel Solver Libraries to solve Structural and Fluid problems", White paper by Cranes Software, http://www.intel.com/cd/ software/products/asmo-na/eng/373466.htm.

[6] Y. Takahashi, 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, pp. 2413–2416, May 2013.

[7] B. He, P. Zhou, C. Lu, N. Chen, D. Lin, B. Bork, "A Time Decomposition Method for the Transient Simulation of Low- Frequency Electromagnetics", presented at the Proc. 17th Biennial IEEE CEFC, Miami, FL, USA, 2016, WO23-5.