Algorithms for Distributed Computation of Electromagnetic Transients towards PC Cluster Based Real-Time Simulations

Taku Noda[1] and Saburo Sasaki[2]

(1) Electrical Insulation Dept., CRIEPI, 2-11-1 Iwado-kita, Komae-shi, Tokyo, 201-8511, Japan (e-mail: takunoda@ieee.org), (2) Komae Research Lab., CRIEPI, 2-11-1 Iwado-kita, Komae-shi, Tokyo, 201-8511, Japan (e-mail: sasaki@criepi.denken.or.jp).

***Abstract* – This paper describes algorithms for the distributed computation of electromagnetic transients in a power system. Conventional simulators based on parallel computing utilize traveling-time delays due to Bergeron's representation of transmission lines in order to divide the network of interest into subnetworks, and each subnetwork is assigned to a single CPU to solve the decoupled sets of the circuit equations in parallel. However, if a large subnetwork which cannot be divided further by a transmission line is found, this limits the overall performance of the simulator. In the case of a real-time simulator, this forces a large time step. To solve this bottleneck, two algorithms are described in this paper. One divides the network at series inductors and the other at shunt capacitors (series and shunt with respect to the ground). Since series inductors and shunt capacitors are found in many locations in a power system, the algorithms increase the possibility for dividing a network into subnetworks achieving optimal computational load assignment to the CPUs in a parallel computer.**

Keywords – algorithms, distributed computing, electromagnetic transient simulations, PC cluster, and real-time simulations.

I. Introduction

High-performance and real-time simulation of electromagnetic transients in a power system requires parallel computing. The nodal conductance formulation that uses the trapezoidal rule of integration for lumped elements and Bergeron's representation for distributed elements [1] has been extended and applied to parallel computing [2–4]. Those parallel simulators utilize the traveling-time delays due to Bergeron's representation of transmission lines in order to divide the network of interest into subnetworks, and each subnetwork is assigned to a single CPU to solve the decoupled sets of the circuit equations in parallel. However, depending on network topology we sometimes find a large subnetwork which cannot be divided further by a transmission line. In this situation, the overall computational performance is limited by the computational speed of the CPU to which the largest subnetwork is assigned. In the case of a real-time simulator, this forces a large time step.

To overcome the bottleneck, this paper describes two algorithms for the distributed computation of electromagnetic transients in a power system. One of the algorithms divides the network of interest at a group of inductors in series with respect to the ground. Each of the inductors is replaced by an equivalent current source whose value is obtained only by past history values of the voltage and the current. The other algorithm divides the network at a group of capacitors in parallel with respect to the ground, and each of the capacitors is replaced by an equivalent voltage source obtained only by past history values. The equivalent voltage and current sources decouple the subnetworks and allow to solve each subnetwork separately in parallel. Since series inductors and shunt capacitors can be found at many locations in a power system, the algorithms increase the possibility for dividing a network into subnetworks which achieve optimal computational load assignment to the CPUs in a parallel computer.

The algorithms are implemented and tested on a PC cluster system consisting of four PCs. Using a 100-BaseTX Ethernet, a possibility of real-time simulation with a time step of 85 μs is confirmed. A cascaded pi circuit is used for the test of the algorithms, and neither error nor numerical instability is observed.

II. Algorithms

A. Central Difference Formula

Let $y(t)$ be the time derivative of $x(t)$:

$$y(t) = \frac{dx(t)}{dt} \tag{1}$$

Both algorithms described in this paper take advantage of the fact that the finite difference of x between $t - \Delta t$ and t is a good approximation of y at $t - \Delta t/2$:

$$y(t - \tfrac{\Delta t}{2}) \cong \frac{x(t) - x(t-\Delta t)}{\Delta t} \tag{2}$$

This is known as the central difference formula.

B. Algorithm 1 – Equivalent Current Source Representation of Series Inductor

The voltage-current relation of an inductor is

$$v(t) = L\frac{di(t)}{dt}\,. \tag{3}$$

Applying (2) to (3), we obtain

$$i(t) = i(t-\Delta t) + \frac{\Delta t}{L} v(t - \tfrac{\Delta t}{2})\,. \tag{4}$$

Since the value of $v(t - \Delta t/2)$ is not available in a discrete time step simulation, it is estimated by the linear approximation using the points $v(t - \Delta t)$ and $v(t - 2\Delta t)$:

$$v(t-\tfrac{\Delta t}{2}) \cong \tfrac{3}{2}v(t-\Delta t)-\tfrac{1}{2}v(t-2\Delta t) \qquad (5)$$

Substituting (5) into (4) gives

$$i(t)=i(t-\Delta t)+\frac{\Delta t}{2L}\{3v(t-\Delta t)-v(t-2\Delta t)\}\,. \qquad (6)$$

This expression indicates that the inductor is now represented by a current source whose value is obtained by the past history values $i(t-\Delta t)$, $v(t-\Delta t)$ and $v(t-2\Delta t)$.

Consider two subnetworks in a power system. The two subnetworks are connected through series inductors as illustrated in Fig. 1 (a). Applying (6) to each inductor, the inductors are replaced by the equivalent current source as in Fig. 1 (b). Since Fig. 1 (c) is equivalent to Fig. 1 (b), finally the two subnetworks are decoupled to each other and the two sets of the circuit equations can be solved separately with two CPUs in parallel.

C. Algorithm 2 – Equivalent Voltage Source Representation of Shunt Capacitor

The current-voltage relation of a capacitor is

$$i(t)=C\frac{dv(t)}{dt}\,. \qquad (7)$$

In the same manner as in the series inductor case, we obtain

$$v(t)=v(t-\Delta t)+\frac{\Delta t}{2C}\{3i(t-\Delta t)-i(t-2\Delta t)\}\,. \qquad (8)$$

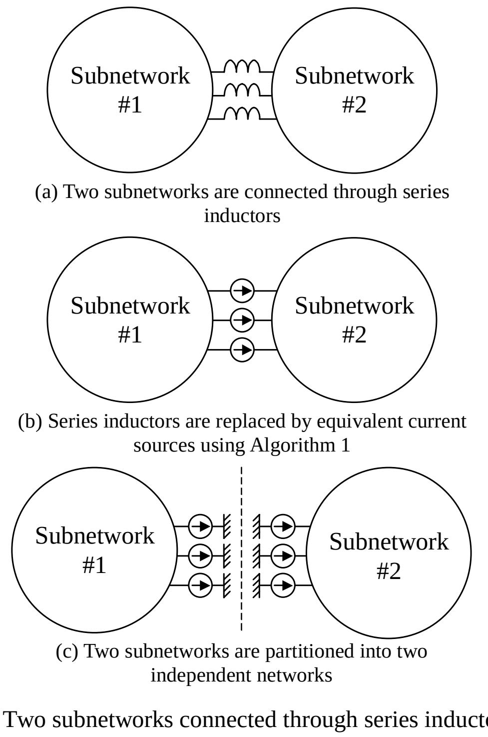
This indicates that the capacitor is represented by a voltage source whose value is obtained by the past history values $v(t-\Delta t)$, $i(t-\Delta t)$ and $i(t-2\Delta t)$.

Fig. 2 (a) shows two subnetworks connected through shunt capacitances. Applying (8) to each of the capacitors gives Fig. 2 (b), where the capacitors are replaced by the equivalent voltage sources. Finally, we obtain Fig. 2 (c), and the two subnetworks are decoupled to each other so that the two sets of the circuit equations can be solved in parallel.

D. Network Partitioning and Simulation Sequence

Fig. 3 illustrates an example of network partitioning using Algorithms 1, 2, and Bergeron's equivalent of a transmission-line. Fig. 3 (a) shows a power network consisting of three subnetworks. Subnetworks #1 and #2 are connected through series inductors, #2 and #3 through shunt capacitors, and #3 and #1 through a transmission line. Algorithm 1 is applied to each of the inductors to decouple Subnetworks #1 and #2, and Algorithm 2 to each of the capacitors to decouple Subnetworks #2 and #3. Each end of the transmission line is represented by current sources accompanied by a multiphase conductance element using Bergeron's representation, and Subnetworks # 3 and #1 are decoupled. Finally, those three subnetworks become independent, and the circuit equations of each subnetwork are assigned to a single CPU to be solved separately in parallel.

Fig. 4 shows the flow chart of the simulation sequence using the proposed algorithms. Fig. 4 is for the case in which the network of interest is partitioned into three subnetworks and solved by three CPUs in parallel as in the above example. At each time step, each CPU solves the circuit equations of the assigned subnetwork. To update the value of the equivalent voltage source of a series inductor used for partitioning, the previous voltage value $v(t-\Delta t)$ across the inductor is required in (6). This indicates that the node voltages at both ends of the inductor at the previous time step have to be exchanged between the two CPUs that

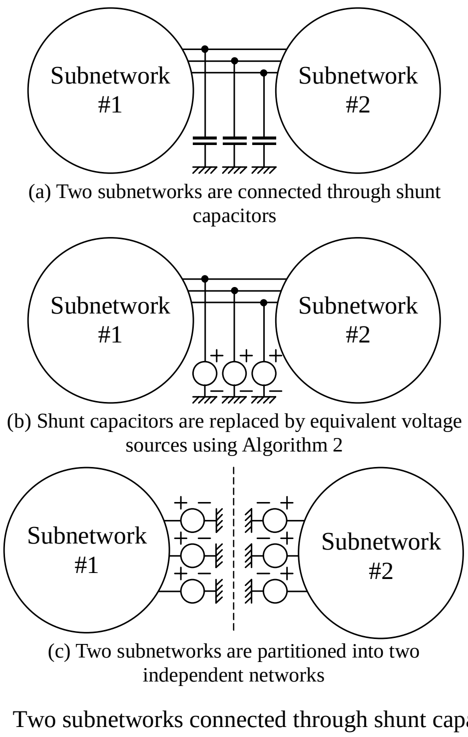


(a) Two subnetworks are connected through series inductors

(b) Series inductors are replaced by equivalent current sources using Algorithm 1

(c) Two subnetworks are partitioned into two independent networks

Fig. 1. Two subnetworks connected through series inductors and their partitioning by Algorithm 1.



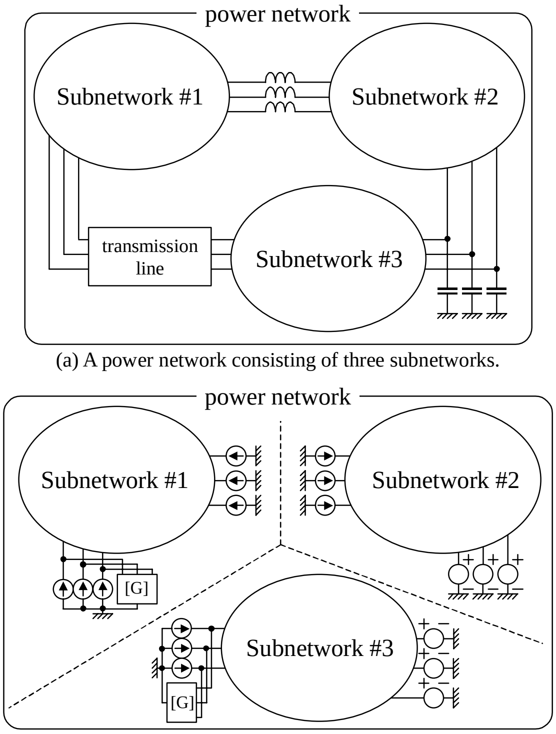
(a) Two subnetworks are connected through shunt capacitors

(b) Shunt capacitors are replaced by equivalent voltage sources using Algorithm 2

(c) Two subnetworks are partitioned into two independent networks

Fig. 2. Two subnetworks connected through shunt capacitors and their partitioning by Algorithm 2.

solve the two subnetworks connected by the inductor. The other values $i(t - \Delta t)$ and $v(t - 2\Delta t)$ appeared in (6) are readily available without inter-CPU communication. In the same way, to update the value of the equivalent current source of a shunt capacitor used for partitioning, the previous current value $i(t - \Delta t)$ through the capacitor is required in (8), and the injected current values to the capacitor from the two connected subnetworks are exchanged between the corresponding CPUs. It should be noted that the inter-CPU communications can be performed in parallel with the past-history update of models, which are not used for partitioning, since most MPI (Message Passing Interface) libraries have functions (subroutines) for non-blocking communications. The functions for non-blocking communications generate a new process for an inter-CPU communication in the corresponding CPUs, and waiting times for the inter-CPU communication can be efficiently used by the past-history update of models.



(a) A power network consisting of three subnetworks.

(b) Subnetworks are decoupled to each other by Algorithms 1, 2, and Bergeron's equivalent of transmission line.

Fig. 3. Example of network partitioning.

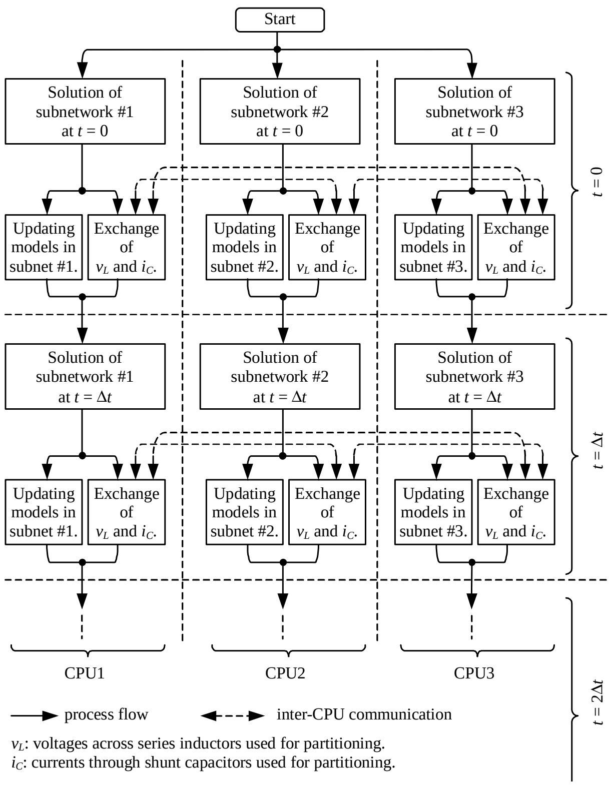


Fig. 4. Simulation sequence.

III. Test Case

A. PC Cluster

A PC cluster system for the distributed simulation of electromagnetic transients was developed at CRIEPI. It consists of four PCs, and each PC has an Intel Pentium 4 (1.8A GHz) processor with an Intel i845 chipset running the RedHat Linux 7.3 operating system. 1 GB RAM and 60 GB HDD are equipped for each PC. Those PCs are connected to each other by a 100-BaseTX Ethernet. The communications among the PCs are carried out by the LAM/MPI library (Local Area Multicomputer / Message Passing Interface: a software library for communications among processors based on the MPI standard) developed at Indiana University. The simulation code for the present paper is written in the C language with LAM/MPI.

B. Test Network

Fig. 5 shows the test network used for the validation of the proposed algorithms. The test network is a cascaded pi

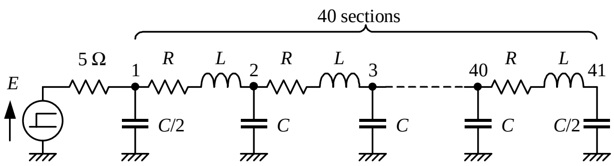
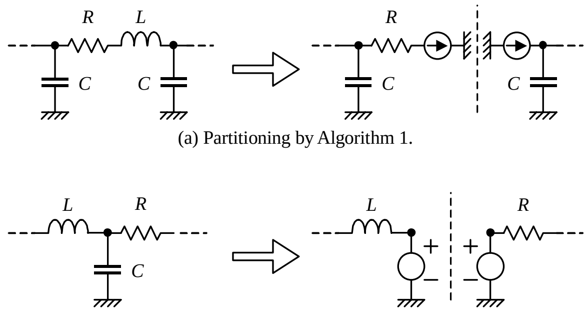


Fig. 5. Test network.



(a) Partitioning by Algorithm 1.

(b) Partitioning by Algorithm 2.

Fig. 6. Partitioning by Algorithms 1 and 2.

circuit of 40 sections with the values R = 6 Ω, L = 912 mH, and C = 4.392 μF. The left-end node is excited by a unit-step voltage through a 5-Ω resistor. From left to right, 10 sections each are assigned to four PCs, i.e., the network is divided into four subnetworks.

Two simulation cases are considered. In Case 1, the series inductors at the boundaries of the subnetworks are replaced by the corresponding equivalent current source by Algorithm 1 as illustrated in Fig. 6 (a). In Case 2, the shunt capacitors at the boundaries are replaced by the corresponding equivalent voltage source by Algorithm 2 as in Fig. 6 (b).

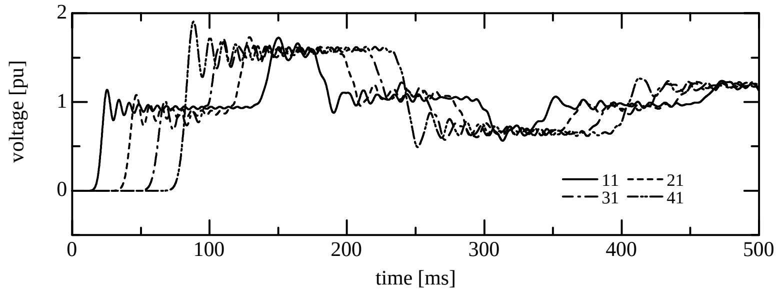
C. *Simulation Result and Performance*

Fig. 7 (a) shows the result obtained by the conventional single CPU computation by EMTP. Figs. 7 (b) and (c) respectively show the results of Cases 1 and 2, obtained by the proposed parallel computation using the PC cluster. In all cases, a time step of 85 μs is used. The voltage waveforms of the nodes at the right-hand ends of the four subnetworks, Nodes 11, 21, 31, and 41, are shown. No difference is observed among the three cases in comparison of the waveforms, and differences can be found only in comparison of tail digits of the calculated numbers. Also, no numerical instability is observed in both Cases 1 and 2. Currently, the PC cluster does not have a synchronization mechanism with real time. However, we have confirmed that the computation times for Cases 1 and 2 are slightly smaller than real time with 85 μs time step.

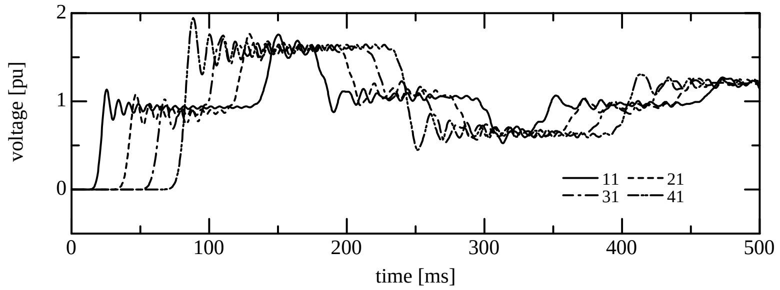
In the test network, the number of nodes assigned to each CPU is 10, and a simulation case with the maximum observation time T_{max} = 1 sec. can be performed in an execution time of 1 sec. To investigate the simulation performance, the number of nodes assigned to each CPU is varied from 10 to 300, and their execution times are plotted in Fig. 8. If it is assumed that the execution time is in proportional to the number of nodes for each CPU, we obtain

$$(\text{execution time}) = 0.1 \times (\text{number of nodes}), \qquad (9)$$

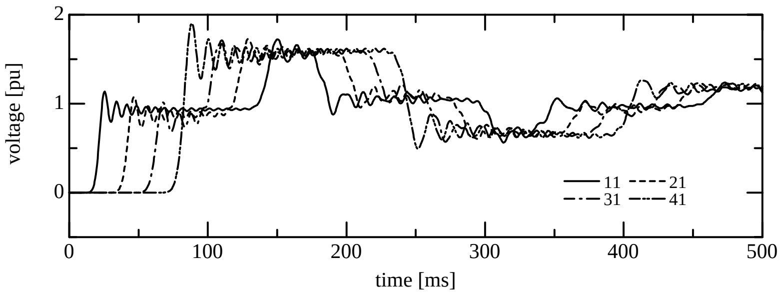
since the execution time for the 10 node case is in real time. Equation (9) is plotted by the broken line in Fig. 8. It is roughly estimated that when the number of nodes for each CPU is small, say a few tens, the computation time for inter-CPU communications is dominant compared with that for solving the circuit equations. Thus, if a faster communication method is used, the performance can be improved.



(a) Result by EMTP



(b) Case1, Result by Algorithm 1



(c) Case 2, Result by Algorithm 2

Fig. 7. Simulation results.

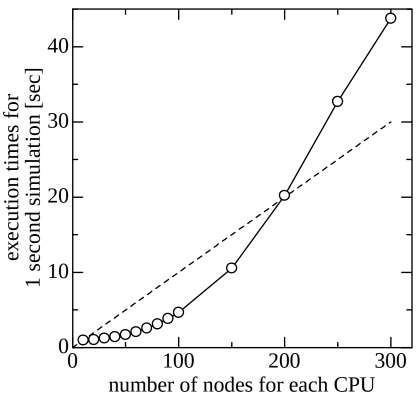


Fig. 8. Variation of execution times for 1 second simulation with respect to number of nodes assigned to each CPU.

Regarding this fact, the authors plan to use a Gigabit Ethernet (1000Base-T) to improve the performance.

III. CONCLUSIONS

This paper has described two algorithms for the distributed computation of electromagnetic transients in a power system. One divides the network of interest at series inductors by replacing them with equivalent current sources. The other divides the network at shunt capacitors by replacing them with equivalent voltage sources. Series inductors and shunt capacitors can be found in many locations in a power system, and thus the algorithms contribute to achieve optimal computational load assignment to the CPUs in a parallel computer.

A cascaded pi circuit is used for the validation of the algorithms, and the computational performance is investigated. A simulation case with the maximum observation time T_{max} = 1 second can be performed in an execution time of 1 second by a PC cluster system developed by the authors. It has been found that when the circuit size for each CPU is small, in the order of a few tens of nodes, the time for inter-CPU communications is dominant compared with that for solving the circuit equations. The authors plan to use a Giga-bit Ethernet to improve the performance of the PC cluster aiming at low-cost real-time simulations.

The application of the presented algorithms to a real-time or parallel computer based simulator is a patent pending technology of CRIEPI.

REFERENCES

[1] H.W. Dommel, “Digital computer solution of electromagnetic transients in single- and multi-phase networks,” *IEEE Trans. on Power App. and Syst.*, Vol. PAS-88 (4), pp. 388-398, 1969.

[2] P.G. McLaren, R. Kuffel, R. Wierckx, J. Giesbrecht, and L. Arendt, “A real time digital simulator for testing relays,” *IEEE Trans. on Power Delivery*, Vol. 7, No. 1, pp. 207-213, 1992.

[3] O. Devaux, L. Levacher, and O. Huet, “An advanced and powerful real-time digital transient network analyser,” *IEEE Trans. on Power Delivery*, Vol. 13, No. 2, pp. 421-426, 1998.

[4] V.-Q. Do, J.-C. Soumagne, G. Sybille, G. Cloutier, and S. Poulin, “HYPERSIM, an integrated real-time simulator for power networks and control systems,” *ICDS ’99*, Vasteras, Sweden, 1999.