A Parallelization-in-time Approach for Accelerating EMT Simulations

Ming Cai, Jean Mahseredjian, Ilhan Kocar, Xiaopeng Fu, Aboutaleb Haddadi

***Abstract*—This paper is related to research on parallelization techniques for the simulation of electromagnetic transients (EMTs). It presents a new approach based on a parallel-in-time equation grouping and reordering (PEGR) technique whose original theoretical formulation is modified, extended and generalized from state-space to the modified-augmented-nodal analysis (MANA) formulation method. The highly efficient sparse linear solver KLU is incorporated in the approach for power system network matrix solution. The proposed approach is implemented with OpenMP Multithreading for CPU-based parallelization. Test results show that EMT simulations can be accelerated by exploiting the intrinsic independency of certain solution procedures using the PEGR technique.**

***Index Terms*--Electromagnetic transients, equation grouping and reordering, parallelization-in-time, OpenMP, KLU**

I. Introduction

Thanks to its wideband nature, the electromagnetic transient (EMT) solution approach has become an indispensable tool for power system engineers to perform a wide range of studies [1][2]. The EMT-type simulation methods are circuit based and usually require significant computing time to solve large differential and algebraic equation (DAE) systems when compared to phasor domain methods due to higher accuracy and model complexity levels. Numerical integration time-step, on the other hand, can also pose a constraint for simulation efficiency even for smaller networks. All of these have prompted the research on computing time reduction for the simulation of EMTs.

Over the years, many techniques have been proposed to improve the solution speed in EMT-type simulation methods, in which parallelization and multistep solution methods have drawn a fair amount of interest among researchers. The former includes approaches using waveform relaxation [3], and techniques implemented in real-time [4]-[6] as well as off-line [7]-[9] simulation tools. In these methods, parallelization is achieved by taking advantage of the propagation delay at distributed parameter transmission line and cable models for natural decoupling of networks. Parallelization can also be realized by identifying intrinsically independent solution procedures using an equation grouping and reordering technique [10]. Multistep solution is attained by exploiting the property of circuit latency [11], or through “data-smoothing” at line-bus interfaces in a network [12]. Although computational speedup has been observed in some cases, the implementation of such multistep techniques on large-scale networks requires user intervention and remains complex to automate. Additionally, a co-simulation-based parallel and multistep approach using the Functional Mock-up Interface standard was proposed in [13].

Other techniques to enhance computational performance in EMT-type solvers include circuit reduction [14], frequency domain fitting [15] and hybrid methods that interface EMT-type solvers with transient stability (TS) programs by exploring parallelism between the two types of solvers [16].

This paper uses an off-line approach based on the parallel-in-time equation grouping and reordering (PEGR) technique for large power system EMT simulations on conventional multi-core computers. This approach automatically groups the network equations at different numbers of solution points, catering to the number of available logical processors on a PC, and recursively applies the PEGR technique (hereinafter labeled as PEGR) on the grouped network equations, thereby significantly reducing the actual number of forward and backward substitution steps in the network equation solution process with the help of multithreading. Special treatment for topological changes and arbitrary numbers of simulation points is also included.

The PEGR was first proposed and theorized in state-space formulation [10]. The approach presented in this paper modifies and extends its original formulation from state-space to the more advantageous modified-augmented-nodal analysis (MANA) formulation method [1]. The highly efficient sparse linear solver KLU [17] is used as the network matrix solver.

This paper starts with the theoretical formulation of PEGR in MANA and follows by its implementation using OpenMP Multithreading. Test cases on power system benchmarks are presented to demonstrate the accuracy and numerical performance advantages of the proposed approach.

II. Formulation of the PEGR Technique in MANA

A. Modified-augmented-nodal Analysis (MANA) and PEGR

The new approach proposed in this paper is implemented based on the modified-augmented-nodal analysis (MANA) formulation method [1]. This method offers several advantages

M. Cai, J. Mahseredjian, I. Kocar and A. Haddadi are with Polytechnique Montreal, Montreal, QC H3C 3A7, Canada (e-mail: jean.mahseredjian@polymtl.ca), X. Fu is with Key Laboratory of Smart Grid of Ministry of Education, Tianjin University, Tianjin, China (e-mail: fuxiaopeng@tju.edu.cn).
Paper submitted to the International Conference on Power Systems Transients (IPST2021) in Belo Horizonte, Brazil June 6-10, 2021.

[1], [18] over classical nodal analysis and state-space formulation. The general MANA equations of a power system network are given by

$$\mathbf{Ax} = \mathbf{b}. \tag{1}$$

The sparse system of equations (1) is solved for $\mathbf{x}$ at each time-point in time-domain, after updating the non-symmetric coefficient matrix $\mathbf{A}$ and the vector of known variables $\mathbf{b}$. The vector $\mathbf{b}$ contains history terms and other independent functions, such as independent sources. The system (1) is non-symmetric and can also accommodate generic equations.

B. Expansion of Network Equations

Since the solutions at two consecutive time-points are not directly formulated into one single equation in the original MANA approach due to the need of updating the history terms of energy storage devices in $\mathbf{b}$ at the solution of each time-point, it is needed to modify PEGR such that it can be adapted into MANA formulation. Taking the multiphase pi-section as an example, using the trapezoidal rule, its discretized vector equations are given by

$$\mathbf{Sv}_{\mathbf{km},t} - \mathbf{i}_{RL,t} = -\mathbf{Sv}_{\mathbf{km},t\text{-}\Delta t} - \mathbf{SHi}_{RL,t\text{-}\Delta t} \tag{2}$$

$$\frac{2}{\Delta t}\mathbf{Cv}_t - \mathbf{i}_{C,t} = \frac{2}{\Delta t}\mathbf{Cv}_{t\text{-}\Delta t} + \mathbf{i}_{C,t\text{-}\Delta t}, \tag{3}$$

where $\mathbf{S} = \left[\mathbf{R} + (2/\Delta t)\mathbf{L}\right]^{\mathbf{-1}}$ and $\mathbf{H} = \left[(2/\Delta t)\mathbf{L} - \mathbf{R}\right]$ and the subscripts $\mathbf{k}$ and $\mathbf{m}$ represent the nodes between which the component is connected.

Suppose the system (1) has n pi-sections. Incorporating these RL current terms $\mathbf{i}_{RL1}$, …, $\mathbf{i}_{RLn}$ and shunt current terms $\mathbf{i}_{C1}$, …, $\mathbf{i}_{Cn}$ into the vector of unknowns $\mathbf{x}$, the expanded network equations (1) can be rewritten as is shown in Figure 1. It is noted that the terms $C_{n,1}$, $C_{n,2}$ represent the capacitances at the two sides of the n^{th} RL section.

$$\overbrace{\begin{bmatrix} & & & \mathbf{A} & & & & \mathbf{0} \\ \cdots & \mathbf{S}_1 & \cdots & -\mathbf{S}_1 & \cdots & \cdots & \cdots & \\ \cdots & \frac{2}{\Delta t}\mathbf{C}_{1,1} & \cdots & \cdots & \cdots & \cdots & \cdots & \\ \cdots & \cdots & \cdots & \frac{2}{\Delta t}\mathbf{C}_{1,2} & \cdots & \cdots & \cdots & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & -\mathbf{I} \\ \cdots & \cdots & \mathbf{S}_n & \cdots & -\mathbf{S}_n & \cdots & \cdots & \\ \cdots & \cdots & \frac{2}{\Delta t}\mathbf{C}_{n,1} & \cdots & \cdots & \cdots & \cdots & \\ \cdots & \cdots & \cdots & \cdots & \frac{2}{\Delta t}\mathbf{C}_{n,2} & \cdots & \cdots & \end{bmatrix}}^{\mathbf{D}} \overbrace{\begin{bmatrix} \mathbf{x}_t \\ \mathbf{i}_{RL1,t} \\ \mathbf{i}_{C1,1,t} \\ \mathbf{i}_{C1,2,t} \\ \vdots \\ \mathbf{i}_{RLn,t} \\ \mathbf{i}_{Cn,1,t} \\ \mathbf{i}_{Cn,2,t} \end{bmatrix}}^{\underline{\mathbf{x}}_t} - \overbrace{\begin{bmatrix} & & & \mathbf{M} & & & & \mathbf{N} \\ \cdots & -\mathbf{S}_1 & \cdots & \mathbf{S}_1 & \cdots & \cdots & \cdots & \\ \cdots & \frac{2}{\Delta t}\mathbf{C}_{1,1} & \cdots & \cdots & \cdots & \cdots & \cdots & \\ \cdots & \cdots & \cdots & \frac{2}{\Delta t}\mathbf{C}_{1,2} & \cdots & \cdots & \cdots & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \mathbf{P} \\ \cdots & \cdots & -\mathbf{S}_n & \cdots & \mathbf{S}_n & \cdots & \cdots & \\ \cdots & \cdots & \frac{2}{\Delta t}\mathbf{C}_{n,1} & \cdots & \cdots & \cdots & \cdots & \\ \cdots & \cdots & \cdots & \cdots & \frac{2}{\Delta t}\mathbf{C}_{n,2} & \cdots & \cdots & \end{bmatrix}}^{\mathbf{O}} \overbrace{\begin{bmatrix} \mathbf{x}_{t-\Delta t} \\ \mathbf{i}_{RL1,t-\Delta t} \\ \mathbf{i}_{C1,1,t-\Delta t} \\ \mathbf{i}_{C1,2,t-\Delta t} \\ \vdots \\ \mathbf{i}_{RLn,t-\Delta t} \\ \mathbf{i}_{Cn,1,t-\Delta t} \\ \mathbf{i}_{Cn,2,t-\Delta t} \end{bmatrix}}^{\underline{\mathbf{x}}_{t\text{-}\Delta t}} = \overbrace{\begin{bmatrix} \mathbf{b}_t \\ \mathbf{0} \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \mathbf{0} \end{bmatrix}}^{\underline{\mathbf{b}}_t}$$

Figure 1. Expanded network equations incorporating current history terms into the vector of unknowns, demonstrating interdependency of solutions at two consecutive time-points.

The vectors $\underline{\mathbf{x}}_t$ and $\underline{\mathbf{b}}_t$ denote the expanded vectors of unknown and known variables (vector $\underline{\mathbf{b}}_t$ is now devoid of history terms) at the time-point t. It is also noted in Figure 1 that the upper left part of matrix $\mathbf{D}$ is the original MANA coefficient matrix $\mathbf{A}$ of the system in (1), the upper right part of $\mathbf{D}$ consists of only zeros and the lower right part of $\mathbf{D}$ is a negative identity matrix. The expanded network equations with individual inductors and capacitors can be derived accordingly.

The equations in Figure 1 depict interdependency between two consecutive time-points t and $t - \Delta t$. They can be written in a compact form

$$\mathbf{D}\underline{\mathbf{x}}_t - \mathbf{O}\underline{\mathbf{x}}_{t\text{-}\Delta t} = \underline{\mathbf{b}}_t, \tag{4}$$

To allow for the implementation of PEGR [10].

C. Grouping and Recursive Row and Column Reordering of Expanded Network Equations

The original PEGR proposition restricts the total number of solution steps T of a simulation to be $T = 2^\tau$ (τ is an integer) in order to solve the simulation in $\log_2 T$ steps using $T/2$ processors, which is unrealistic in practice. Hence, instead of grouping the solutions at all time-points to construct a huge matrix as in the original proposition, this section presents the formulation of the new approach with an example showing the grouping of expanded network equations at 16 consecutive time-points, thus solving the network for every 16 time-steps. The algorithm also accommodates the solution of every 32, 64, …, T time-steps ($T = 2^{\tau+5}, \tau = 0,1,2,\ldots$).

Defining $\hat{\underline{\mathbf{x}}} = \left[\underline{\mathbf{x}}_1,\ldots,\underline{\mathbf{x}}_{16}\right]$ and $\hat{\underline{\mathbf{b}}} = \left[\underline{\mathbf{b}}_1,\ldots,\underline{\mathbf{b}}_{16}\right]$ a large set of the expanded network equations consisting of solutions at the first 16 time-points can therefore be constructed:

$$\begin{bmatrix} \mathbf{D} & & & & \\ \mathbf{O} & \mathbf{D} & & & \\ & \mathbf{O} & \mathbf{D} & & \\ & & \ddots & \ddots & \\ & & & \mathbf{O} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{x}}_1 \\ \underline{\mathbf{x}}_2 \\ \underline{\mathbf{x}}_3 \\ \vdots \\ \underline{\mathbf{x}}_{16} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{b}}_1 \\ \underline{\mathbf{b}}_2 \\ \underline{\mathbf{b}}_3 \\ \vdots \\ \underline{\mathbf{b}}_{16} \end{bmatrix} \tag{5}$$

It is noted that initialization data from steady-state or load-flow are included in $\underline{\mathbf{b}}_1$. The above network equations can also be written in a compact manner

$$\hat{\mathbf{A}}\underline{\hat{\mathbf{x}}} = \underline{\hat{\mathbf{b}}}. \tag{6}$$

After performing a one-time row and column reordering on $\hat{\mathbf{A}}$ by renumbering all odd rows and columns ahead of all even rows and columns and applying an LU factorization technique:

$$\hat{\mathbf{A}}\underline{\hat{\mathbf{x}}} = \mathbf{LU}\underline{\hat{\mathbf{x}}} = \mathbf{L}\underline{\hat{\mathbf{y}}} = \underline{\hat{\mathbf{b}}} \tag{7}$$

The lower triangular equations in the L side for forward substitution to solve for $\underline{\hat{\mathbf{y}}}$ are given by

$$\left[\begin{array}{ccccc|ccccc}
\mathbf{d} &&&&&&&&& \\
& \mathbf{d} &&&&&&&& \\
&& \ddots &&&&&&& \\
&&& \mathbf{d} &&&&&& \\
&&&& \mathbf{d} &&&&& \\
\hline
\mathbf{o} &&&&& \mathbf{d} &&&& \\
& \mathbf{o} &&&& \mathbf{f} & \mathbf{d} &&& \\
&& \ddots &&&& \mathbf{f} & \ddots && \\
&&& \mathbf{o} &&&& \ddots & \mathbf{d} & \\
&&&& \mathbf{o} &&&& \mathbf{f} & \mathbf{d}
\end{array}\right]
\left[\begin{array}{c}
\underline{\mathbf{y}}_1 \\ \underline{\mathbf{y}}_3 \\ \vdots \\ \underline{\mathbf{y}}_{13} \\ \underline{\mathbf{y}}_{15} \\ \hline \underline{\mathbf{y}}_2 \\ \underline{\mathbf{y}}_4 \\ \vdots \\ \underline{\mathbf{y}}_{14} \\ \underline{\mathbf{y}}_{16}
\end{array}\right]
=
\left[\begin{array}{c}
\underline{\mathbf{b}}_1 \\ \underline{\mathbf{b}}_3 \\ \vdots \\ \underline{\mathbf{b}}_{13} \\ \underline{\mathbf{b}}_{15} \\ \hline \underline{\mathbf{b}}_2 \\ \underline{\mathbf{b}}_4 \\ \vdots \\ \underline{\mathbf{b}}_{14} \\ \underline{\mathbf{b}}_{16}
\end{array}\right] \tag{8}$$

The $\mathbf{d}$, $\mathbf{o}$ and $\mathbf{f}$ represent diagonal, off-diagonal and fill-in blocks respectively (the numerical values of the same blocks are not necessarily identical). It is easy to observe that the first 8 equations in (8) become independent and can be solved in parallel. However, the remaining 8 equations are still interdependent. Performing the same row and column reordering scheme recursively first on the entire 16 equations, then on the remaining 8, 4 and 2 equations, after LU factorization, gives:

$$\overbrace{\left[\begin{array}{cccccccccccccccc}
\mathbf{d} &&&&&&&&&&&&&&& \\
& \mathbf{d} &&&&&&&&&&&&&& \\
&& \mathbf{d} &&&&&&&&&&&&& \\
&&& \mathbf{d} &&&&&&&&&&&& \\
&&&& \mathbf{d} &&&&&&&&&&& \\
&&&&& \mathbf{d} &&&&&&&&&& \\
&&&&&& \mathbf{d} &&&&&&&&& \\
&&&&&&& \mathbf{d} &&&&&&&& \\
\mathbf{o} &&&&&&&& \mathbf{d} &&&&&&& \\
&& \mathbf{o} &&&&&&& \mathbf{d} &&&&&& \\
&&&& \mathbf{o} &&&&&& \mathbf{d} &&&&& \\
&&&&&& \mathbf{o} &&&&& \mathbf{d} &&&& \\
& \mathbf{o} &&&&&&& \mathbf{f} &&&& \mathbf{d} &&& \\
&&&&& \mathbf{o} &&&&& \mathbf{f} &&& \mathbf{d} && \\
&&& \mathbf{o} &&&&&& \mathbf{f} &&& \mathbf{f} && \mathbf{d} & \\
&&&&&&& \mathbf{o} &&&& \mathbf{f} && \mathbf{f} & \mathbf{f} & \mathbf{d}
\end{array}\right]}^{\mathrm{L}}
\left[\begin{array}{c}
\underline{\mathbf{y}}_1 \\ \underline{\mathbf{y}}_3 \\ \underline{\mathbf{y}}_5 \\ \underline{\mathbf{y}}_7 \\ \underline{\mathbf{y}}_9 \\ \underline{\mathbf{y}}_{11} \\ \underline{\mathbf{y}}_{13} \\ \underline{\mathbf{y}}_{15} \\ \underline{\mathbf{y}}_2 \\ \underline{\mathbf{y}}_6 \\ \underline{\mathbf{y}}_{10} \\ \underline{\mathbf{y}}_{14} \\ \underline{\mathbf{y}}_4 \\ \underline{\mathbf{y}}_{12} \\ \underline{\mathbf{y}}_8 \\ \underline{\mathbf{y}}_{16}
\end{array}\right]
=
\left[\begin{array}{c}
\underline{\mathbf{b}}_1 \\ \underline{\mathbf{b}}_3 \\ \underline{\mathbf{b}}_5 \\ \underline{\mathbf{b}}_7 \\ \underline{\mathbf{b}}_9 \\ \underline{\mathbf{b}}_{11} \\ \underline{\mathbf{b}}_{13} \\ \underline{\mathbf{b}}_{15} \\ \underline{\mathbf{b}}_2 \\ \underline{\mathbf{b}}_6 \\ \underline{\mathbf{b}}_{10} \\ \underline{\mathbf{b}}_{14} \\ \underline{\mathbf{b}}_4 \\ \underline{\mathbf{b}}_{12} \\ \underline{\mathbf{b}}_8 \\ \underline{\mathbf{b}}_{16}
\end{array}\right] \tag{9}$$

It is observed that the original 16-step forward substitution procedure can now be achieved in 3 parallel steps and 2 sequential steps (in total 5). Following this methodology, the total number of backward substitution steps is reduced from 16 to 4. Therefore, in this example, the number of steps in forward and backward substitutions is reduced from 32 in the conventional step-by-step approach to 9.

Without topological changes in the network or iterations, the matrix $\hat{\mathbf{A}}$ in equation (6) remains unchanged. Major computation time gains can thus be expected in a time-domain simulation scenario. The proposed approach can be applied to group and reorder network equations of every 32, 64, …, T ($T = 2^{\tau+5}, \tau = 0,1,2,...$) time-steps in the exact same fashion using 16, 32, …, S ($S = 2^{\tau+4}, \tau = 0,1,2,...$) processors.

III. Algorithm of the PEGR Approach for Time-domain Simulations

The PEGR presented in this paper is implemented for sparse matrices and with OpenMP Multithreading [19]. The high-performance sparse matrix solver KLU [17] is used as the network matrix solver. The expanded network equations are formulated and grouped based on the number of available logical processors on the PC to achieve maximal parallelism. The computational burden on each parallel thread is optimized in terms of maximizing numerical performance and minimizing concurrency issues. Special measures are also taken in order to minimize false sharing, data race, as well as inconsistency between temporary view and memory in OpenMP.

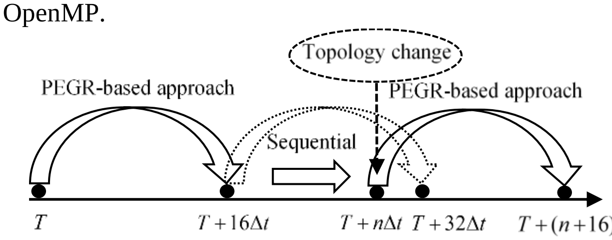


Figure 2. Switching between the PEGR and the conventional approach in the case of topological changes in the network.

Since PEGR groups the expanded network equations at a certain number of time-points and solves them together, a conventional sequential step-wise approach also employing the KLU solver to solve the original network MANA equations is hence incorporated in the algorithm to handle topological changes that happen at arbitrary time-points. In the example of grouping the solutions at 16 time-points, the PEGR is used to solve equations between $t = T$ and $t = T + 16\Delta t$ (Figure 2). Nonetheless, a topological change happens at $t = T + n\Delta t$ which is before $t = T + 32\Delta t$. The algorithm therefore switches to the conventional sequential step-wise approach to solve the steps between $t = T + 16\Delta t$ and $t = T + n\Delta t$, and switches back to the PEGR afterwards.

IV. Test Cases

The new approach proposed in this paper is validated in terms of accuracy and computation time gains. Accuracy is verified by comparing various waveforms using the proposed approach with those obtained from an EMT-type solver [1]. The computation time gains are based on the comparison of the solution times of benchmarks simulated using the proposed approach and the conventional sequential step-wise method, in which both methods employ the KLU solver as the network matrix solver. All tests are executed on a PC with 24 cores and 48 logical processors. In all test cases, a simulation interval of 3s is chosen, and a phase-a-to-ground fault scenario occurring at $t = 2s$ and cleared at $t = 2.1s$ is simulated in time-domain. A numerical integration time-step of $\Delta t = 50\mu s$ is used in all test cases.

A. Accuracy Validation

The Network-1 benchmark, as is illustrated in Figure 3, is used in the test, with the phase-a-to-ground fault occurring at bus 890. It is based on the original IEEE-34 test system [20]. In this test, the generator is represented by a three-phase voltage source; the transformers and regulators are modeled by three-phase coupled windings; all the transmission lines are modeled by coupled pi-sections; and the loads are represented using RL branches. The PEGR groups the expanded network equations at 16 consecutive time-points using 8 logical processors. This is adequate for accuracy validation in that the formulation of the expanded network equations before grouping and reordering remains the same despite the number of launched threads and the number of solution time-points at which the expanded network equations are to be grouped.

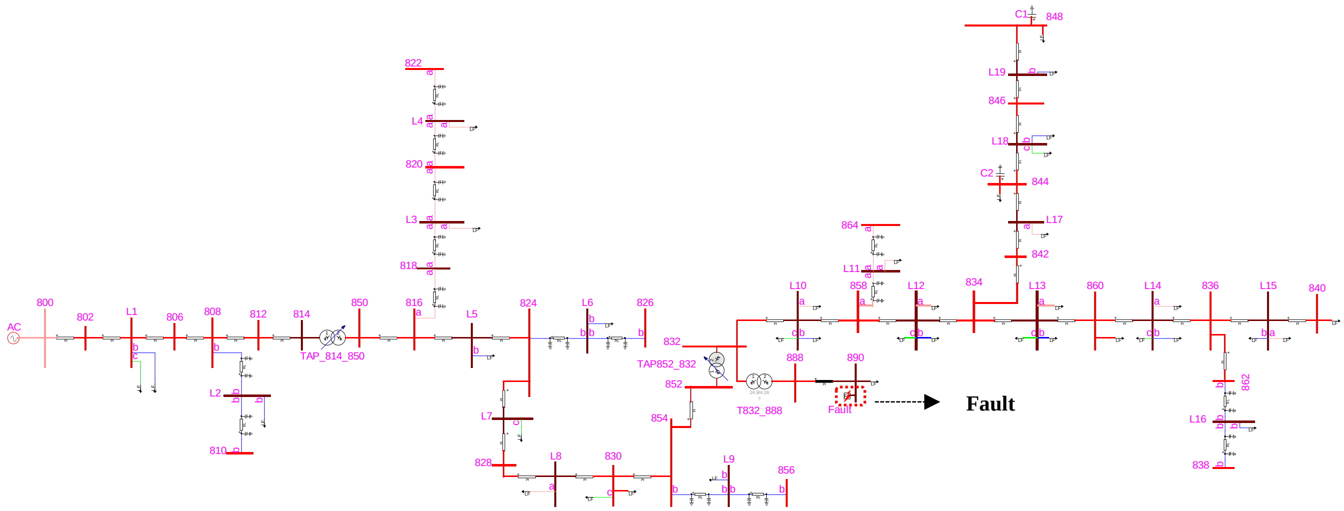
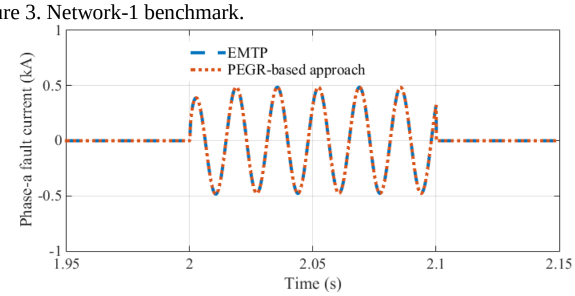
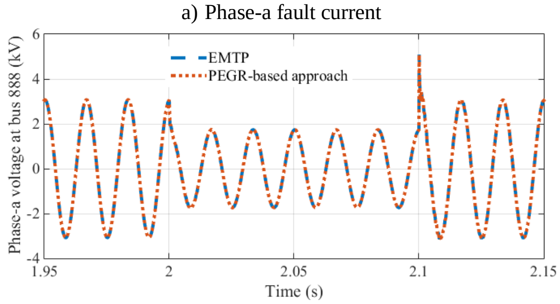


Figure 3. Network-1 benchmark.



a) Phase-a fault current



b) Phase-a voltage at bus 888

Figure 4. Waveforms of the phase-a fault current and phase-a voltage at bus 888.

The waveforms of the phase-a fault current, as well as phase-a voltage at bus 888, are presented in Figure 4. It can be observed that the various waveforms obtained from the proposed PEGR are identical to those from EMTP, demonstrating that time-domain simulations using the proposed approach remain accurate.

B. Computation Time Gains

A 24-core PC, whose hardware and software configurations are given in Table I, is used in the tests. It is worth noting that the purpose here is not to compare the performance of the PEGR simulation algorithm developed in this work with that of any highly optimized commercial power system simulation packages, but to demonstrate the numerical advantages of the new methodology brought by grouping and recursive row and column reordering of expanded network equations over the conventional sequential step-wise solution scheme. Hence, the solution times obtained from PEGR are compared with a conventional sequential step-wise solution scheme implemented on the same programming platform using the KLU sparse linear solver as the matrix solver, and the performance timings using PEGR presented in this section might still be inferior to commercial power system simulation packages with highly optimized routines such as [1] for the tested benchmarks.

TABLE I HARDWARE AND SOFTWARE CONFIGURATIONS OF THE 24-CORE PC

Processor	Intel (R) Xeon (R) E5-2650 v4
# of cores	24
# of logical processors	48
Installed memory (RAM)	32.0 G
Base speed	2.2 GHz

1) Test Case-1

The first test case is based on Network-1 shown in Figure 3. It is performed on the original Network-1 benchmark which is a 34-bus network as well as on a 170-bus network constructed by replicating the Network-1 benchmark 5 times. Different numbers of logical processors (threads) 8, 16, and 32 are used in the tests with the proposed PEGR method. The expanded network equations are thus grouped for every 16, 32, and 64 consecutive time-steps. The performance timings of both approaches are presented in Table II.

TABLE II PERFORMANCE TIMINGS OF TEST CASE-1

	Step-wise method	**PEGR method**		
Number of threads	N/A	8	16	32
34-bus solution time (s)	83.72	50.43	39.03	28.20
170-bus solution time (s)	444.53	233.16	170.45	125.53

Based on the performance timings presented in Table II, the total solution time speedup using the PEGR compared to the conventional sequential step-wise approach with respect to the number of launched threads, is shown in Figure 5 for both test networks.

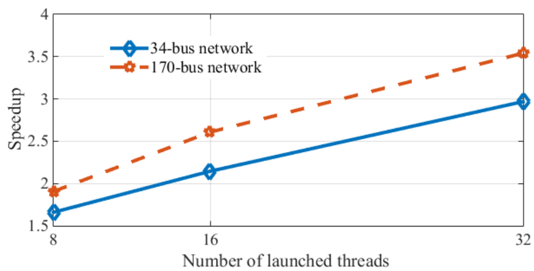


Figure 5. Solution time speedup using the PEGR with respect to the number of launched threads in Case-1.

2) *Test Case-2*

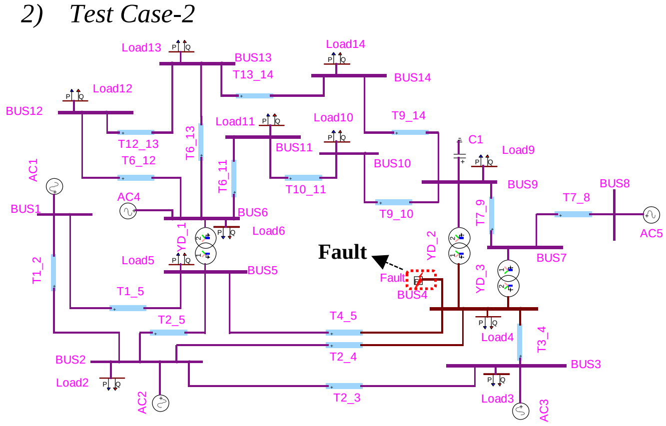


Figure 6. Network-2 benchmark.

The second test case is based on the Network-2 benchmark shown in Figure 6. It is the EMT version of the original IEEE-14 test system. The parameters for this benchmark can be found in [21]. The models are similar to test Case-1. By replicating the Network-2 benchmark 5, 10, 15 and 20 times, larger systems of 70, 140, 210 and 280 buses can therefore be constructed. Once again, different numbers of threads 8, 16, and 32 are used in the tests with the PEGR. Table III presents the performance timings of all test networks using both approaches.

TABLE III PERFORMANCE TIMINGS OF TEST CASE 2

	Step-wise method	**PEGR method**		
Number of launched threads	N/A	8	16	32
14-bus solution time (s)	9.56	12.23	8.65	5.39
70-bus solution time (s)	35.24	21.83	17.40	12.70
140-bus solution time (s)	85.61	42.44	34.98	27.23
210-bus solution time (s)	137.98	59.62	47.89	37.41
280-bus solution time (s)	185.01	72.94	57.89	46.17

Based on the performance timings presented in Table III, the total solution time speedup using the PEGR compared to the conventional sequential step-wise approach with respect to the number of launched threads can be calculated and is shown in Figure 7 for all test networks.

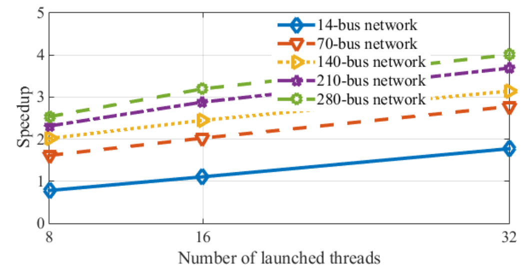


Figure 7. Solution time speedup using the PEGR with respect to the number of launched threads in test Case-2.

From Figure 5 and Figure 7, several conclusions can be drawn as follows. Firstly, it is observed in Figure 5 and Figure 7 that despite the need of operating on a much larger sparse matrix comprised of solutions at 16, 32, and 64 consecutive time-points (as in the cases of using 8, 16, and 32 threads respectively), the PEGR still demonstrates a higher efficiency compared to the conventional step-wise solution scheme in most scenarios for both test cases (except for the 14-bus network using 8 threads (Figure 7).

Secondly, the more threads a simulation launches, the more the forward and backward substitution steps can be parallelized, which leads to a higher speedup for the same test benchmark (see Figure 5 and Figure 7).

Thirdly, it is also observed that the actual speedup obtained in both test cases is lower than that theorized in the original proposition of the PEGR [10]. This is because the original proposition only considers the number of steps the forward and backward substitutions can be theoretically reduced to with sufficient parallel processors whilst not accounting for other procedures required in time-domain simulations, such as equation grouping, recursive row and column reordering, factorization and refactorization of the reordered and grouped network equations, and block element extraction, as well as the overhead brought by the measures taken to minimize concurrency issues in parallel programming and OpenMP inner mechanism.

Lastly, the computational efficiency enhancement using the PEGR for smaller networks can be dwarfed by the time consumed in the supplementary operations in the PEGR that do not exist in the conventional step-wise solution scheme (see the case of the 14-bus network with 8 threads in Figure 7). However, as the network grows larger, the time consumed in the supplementary operations in the PEGR becomes less significant compared to the total solution time, the computational efficiency enhancement brought by the PEGR therefore becomes more remarkable.

Overall, it can be concluded that the proposed PEGR approach can accelerate the time-domain simulations of large power systems as the advantages of parallelism, network equation grouping and recursive row and column reordering scheme being fully exploited.

V. CONCLUSION

This paper experimented with a new approach for acceleration of computations in the simulation of electromagnetic transients. Stemming from a parallel-in-time equation grouping and reordering technique (PEGR), this new approach modifies and extends its original theoretical formulation in state-space to accommodate the modified-augmented-nodal analysis formulation method.

The accuracy and computation time gains of this new approach are demonstrated using power system benchmarks of different levels of complexity as well as those constructed with their multiple replicas while simulating a phase-to-ground fault scenario, in which it is observed that accuracy is properly maintained using the new PEGR approach and satisfactory solution time speedup can be achieved.

The parallel-in-time approach proposed in this paper is the first attempt at adapting the PEGR technique into MANA formulation and implementing it in solving realistic power system networks. Due to its special network equation formulation scheme that differs significantly from conventional EMT step-wise approaches, the proposed approach sheds remarkable insight on fast power system EMT simulations from a different perspective and establishes the base of future work in the possible integration with other techniques to solve for power system dynamic problems on a much larger scale in a potentially multi-step simulation environment.

REFERENCES

[1] J. Mahseredjian, S. Dennetière, L. Dubé, B. Khodabakhchian, L. Gérin-Lajoie, "On a new approach for the simulation of transients in power systems," *Electric power systems research*, vol. 77, no. 11, pp: 1514-1520, 2007.

[2] Ametani, “Numerical Analysis of Power System Transients and Dynamics”, IET (The Institution of Engineering and Technology), 2015.

[3] J. M. Bahi, K. Rhofir, and J.-C. Miellou, "Parallel solution of linear DAEs by multisplitting waveform relaxation methods," *Elsevier, Linear Algebra and its Applications*, vol. 332-334, pp. 181-196, Aug. 2001.

[4] D. Paré, G. Turmel, J.-C. Soumagne, V. Q. Do, S. Casoria, M. Bissonnette, B. Marcoux, and D. McNabb, “Validation tests of the Hypersim digital real time simulator with a large AC-DC network,” in *International conference on power system transients (IPST)*, New Orleans, USA, 2003.

[5] S. Abourida, C. Dufour, J. Belanger, G. Murere, N. Lechevin, and B. Yu, “Real-time PC-based simulator of electric systems and drives,” in *Proc. 17th IEEE APEC, Applied Power Electronics Conf. and Expo*., pp. 433-438, Dallas, USA, Mar. 10-14, 2002.

[6] R. Kuffel, J. Giesbrecht, T. Maguire, R. P.Wierckx, and P. G. McLaren, “RTDS-A fully digital power system simulator operating in real-time,” in *Proc. WESCANEX*, vol.2, pp. 300-305, Winnipeg, Manitoba, Canada, 1995.

[7] R. Singh, A. M. Gole, C. Muller, P. Graham, R. Jayasinghe, B. Jayasekera, and D. Muthumuni, “Using Local Grid and Multi-core Computing in Electromagnetic Transients Simulation,” in *International conference on power system transients (IPST)*, Vancouver, Canada, 2013.

[8] S. Montplaisir-Goncalves, J. Mahseredjian, O. Saad, X. Legrand, and A. El-Akoum, “A Semaphore-based Parallelization of Networks for Electromagnetic Transients,” in *International conference on power system transients (IPST)*, Vancouver, Canada, 2013.

[9] Abusalah, O. Saad, J. Mahseredjian, U. Karaagac, I. Kocar, “CPU based parallel computation of electromagnetic transients for large power grids”, Electric Power Systems Research, Volume 160, July 2018, pp. 57-63.

[10] F. L. Alvarado, "Parallel solution of transient problems by trapezoidal integration,” *IEEE Trans. Power App. Syst.*, vol. PAS-98, no. 3, pp. 1080-1090, May/June 1979.

[11] Benigni, A. Monti, and R. A. Dougal, “Latency-based approach to the simulation of large power electronics systems,” *IEEE Transactions on Power Electronics*, vol. 29, no. 6, pp. 3201–3213, June 2014.

[12] A. Semlyen, and F. De Leon, "Computation of electromagnetic transients using dual or multiple time steps," *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 1274-1281, Aug. 1993.

[13] M. Cai, H. Gras, J. Mahseredjian, E. Rutovic, and A. El-Akoum, “Functional Mock-up Interface-Based Approach for Parallel and Multistep Simulation of Electromagnetic Transients, ” *IEEE Transactions on Power Delivery*, vol. 33, no. 6, pp. 2978-2988, Dec. 2018.

[14] U. N. Gnanarathna, A. M. Gole and R. P. Jayasinghe, “Efficient Modeling of Modular Multilevel HVDC Converters (MMC) on Electromagnetic Transient Simulation Programs,” IEEE Trans. on Power Delivery, vol. 26, no. 1, pp. 316-324, Jan. 2011.

[15] J. Morales, J. Mahseredjian, K. Sheshyekani, A. Ramirez, E. Medina, “Pole-Selective Residue Perturbation Technique for Passivity Enforcement of FDNEs”, IEEE Transactions on Power Delivery, Vol. 33, No. 6, pp. 2746-2754, 2018.

[16] IEEE Task Force on Interfacing Techniques for Simulation Tools, “Interfacing Techniques for Transient Stability and Electromagnetic Transient Programs,” *IEEE Trans. on Power Delivery*, vol. 24, no. 4, pp. 2385-2395, Oct. 2009.

[17] T. A. Davis, and E. P. Natarajan, ”Algorithm 907: KLU, a direct sparse solver for circuit simulations problems,” ACM Trans. Math. Softw., Vol. 37, pp 36:1-36:17, September 2010.

[18] J. Mahseredjian, U. Karaagac, S. Dennetiere, H. Saad, “Simulation of electromagnetic transients with EMTP-RV”, Book, A. Ametani (Editor), “Numerical Analysis of Power System Transients and Dynamics”, IET (The Institution of Engineering and Technology), 2015.

[19] OpenMP user manual, available online: http://www.openmp.org/wp-content/uploads/openmp-4.5.pdf

[20] W. H. Kersting, “Radial distribution test feeders,” IEEE Trans. Power Syst., vol. 6, pp. 975–985, Aug. 1991.

[21] Kamel, S., M. Kodsi, and C.A. Canizares, Modeling and simulation of IEEE 14 bus system with facts controllers. IEEE Student Member–2003, 2009.