A Method for Parallelized Fast Dynamic Cascading Failure Simulation of Power System

Sina Gharebaghi, Nilanjan Ray Chaudhuri, Ting He, and Thomas La Porta
School of Electrical Engineering and Computer Science
The Pennsylvania State University
University Park, PA 16802, USA
svg5765@psu.edu, nuc88@engr.psu.edu, tzh58@psu.edu, and tfl12@psu.edu

***Abstract*—A new approach called Backward Euler method with Predictor-Corrector (BEM-PC) for simulating cascading failure in dynamic models of power systems was recently proposed. It applied Backward Euler integration method (BEM) with stiff decay property while overcoming its so-called hyperstability issue. The method led to a significant simulation speedup without sacrificing accuracy in tracking cascading path when compared with traditional solution techniques like Trapezoidal integration method (TM). In this paper, we demonstrate that a further speedup can be achieved by a parallelized version of BEM-PC, which we call BEM-PC-parallel (BEM-PCP). In this version, the predictor subprocess of BEM-PC is run in multiple parallel processors for identification of oscillatory instability using eigen-decomposition of the system matrix at post-disturbance unstable equilibria. Monte-Carlo studies on a $2,383$-bus Polish system confirm that BEM-PCP is on average 17% faster than BEM-PC and ≈ 40 times as fast as TM while maintaining the same accuracy as BEM-PC.**

***Index Terms*—Dynamic model, Cascading failure, Backward Euler method, Oscillatory instability, Parallel computation, Trapezoidal method.**

I. Introduction

CASCADING failures in power systems can originate from multiple reasons including line tripping and/or failure of primary equipment like generators and transformers. Therefore, it is important to perform statistical analysis of initial outages using a detailed dynamic model of the grid, which can closely represent the ground truth. Unfortunately, this entails repeatedly solving nonlinear coupled differential algebraic equations (DAEs) for a much longer time than typical planning studies that run for $\approx$ 20–30 s. Due to computational complexity, such studies have been proven to be elusive, and have forced the power community to use Quasi-Steady-State (DC-QSS, AC-QSS) models [1], which despite recent improvements [2], cannot capture many phenomena in the dynamic models.

A. Literature Review: Dynamic Models for Cascading Failure

Although some papers have reported research on dynamic cascading failure models, their computational inefficiency issue has largely remained unresolved. Reference [3] has focused on understanding the interaction of protection schemes and system dynamics during cascading outages. In this regard, it provides a brief overview of dynamics and protection, existing modeling techniques, simulation frameworks, and research gaps. A dynamic model of cascading outages in power grids with renewable resources is proposed in [4]. Authors in [5] discusses development of a dynamic model that fits high performance computing simulation environments. Paper [6] suggests a parallel strategy suitable for deployment on a super-computer in order to improve the computational speed of cascading failure simulations. Papers [3]–[6] are review and proposition oriented – they did not perform any comprehensive cascading failure simulations.

Authors in [7] suggests a network-based structure-preserving dynamic model for cascade simulations. The authors attempted to demonstrate that simpler models may not be reliable for investigating cascade outages, and presence or absence of protection mechanism can lead to different results. However, the proposed model is built on the classical model of generators with swing equations, which is not able to capture the stability issues coming from excitation systems [8], [9].

Pacific Northwest National Laboratory's (PNNL's) Dynamic Contingency Analysis Tool (DCAT) [10] simulates severe contingencies by alternating between dynamic and QSS models for cascading failures while integrating the hybrid model with protection schemes and automatic corrective actions. Paper [11] introduces a bi-level probabilistic risk assessment dynamic model for cascading events incorporating slow and fast cascades. A detailed dynamic model, COSMIC, for cascading outage simulation triggered by $N-2$ contingencies has been proposed in [12]. Authors in [13] demonstrate results of a detailed dynamic cascading failure model on a $2,000$-bus test system. A selection of $N-2$ line outages out of the 100 most important lines in the system are applied as the initial disturbance.

B. Gaps in Literature

Although valuable contributions have been made in [10]–[13], the computational cost vs accuracy trade-off of these dynamic models still remains an obstacle for statistical analyses. For example, **(i)** ref. [10]: switching between dynamic and QSS models is complicated, and may lead to inaccuracies in the proposed hybrid model; **(ii)** ref. [12]: COSMIC is effectively tested only for 88 out of the $1,200$ $N-2$ contingencies, which led to dependent failures; **(iii)** ref. [13]: only those

Financial support from NSF Grant Award ECCS 1836827 is gratefully acknowledged.

TABLE I: Performance comparison of the state-of-art cascading failure models

Attributes	Quasi-Steady-State (QSS) Models		Detailed Phasor-based Dynamic Models		
	DC-QSS	AC-QSS	Conventional method (TM/R-K)	BEM-PC	**BEM-PCP**
Accuracy	Highly inaccurate results	Mostly inaccurate results	Ground truth*	∼Ground truth	**Same as BEM-PC**
Simulation speed	Extremely fast	Very fast	Extremely slow	Fast–On average, up to 35 times faster than TM in Polish system	**Faster than BEM-PC**
Statistical analysis	Feasible	Feasible	Not feasible	Feasible	**Feasible**

* We define the 'ground truth' as the cascading failure simulation result produced by the state-of-the-art TM/R-K approaches.

contingencies that stopped in a 50 s simulation time have been analyzed using a constant integration time-step of ≈ 0.004 s. Therefore, these approaches do not solve the computational challenge facing statistical analysis of cascading failures.

C. Overview of BEM-PC Method

A new approach called Backward Euler method with Predictor-Corrector (BEM-PC) [14], [15] was recently proposed by the authors to fill the existing gaps. BEM-PC utilizes Backward Euler integration method (BEM) with stiff decay, which allows large integration step-sizes and achieves a significant simulation speedup compared to traditional Trapezoidal method (TM). The so-called hyperstability issue in BEM is addressed in BEM-PC by introducing a novel predictor-corrector approach.

Hyperstability in an integration method is defined as its property of producing a stable response converging to the unstable equilibrium when solving differential equations with instability. Hyperstability in BEM corresponds to the zone highlighted in gray outside the unit circle in the right-half plane of Fig. 1, while that is not the case for TM. Figure 2 shows time-domain plots of rotor angle of the generator in a single-machine-infinite-bus (SMIB) system following a fault in one of the parallel lines and its tripping at $t = 5$ s. The left subplot in Fig. 2 indicates that for the stable case, BEM with stiff decay can take large time steps and capture identical end results as TM. However, for a case with oscillatory instability (right subplot) BEM converges to an unstable equilibrium, producing erroneous end results. Note that BEM in this figure does not refer to BEM-PC. This reveals that BEM without predictor-corrector approach can be vulnerable in presence of oscillatory instability while it is well-known that oscillatory instability is manifested in many power systems – see [8], [9] for example. In contrast, Monte-Carlo simulation of BEM-PC proposed in [14], [15] indicates a remarkable speedup compared with TM-based simulation while attaining the same cascade paths and end results of cascade with a very high accuracy. This makes BEM-PC a capable tool for exhaustive statistical analysis of cascading failure.

D. Contribution of this Paper

The objective of this paper is to perform the parallelization of a subprocess within BEM-PC, leading to an improved version called BEM-PC Parallel (BEM-PCP). Monte-Carlo simulation of $2,383$-bus Polish system in [14] shows that the predictor subprocess of BEM-PC consumes on average $\approx 25\%$ of total CPU-time for simulating each contingency.

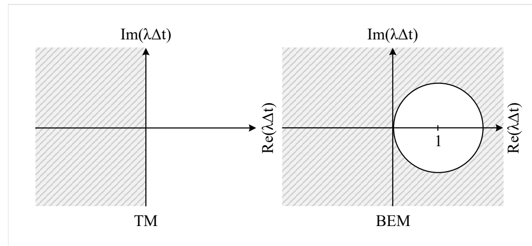

Fig. 1: Absolute stability regions of TM (left) and BEM (right) shown in gray for so-called *test equation* $\dot{x} = \lambda x$. λ is a complex number indicating the eigenvalue of a system matrix.

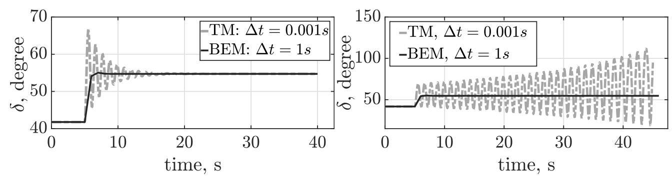

Fig. 2: Rotor angle time-domain plot of SMIB system with a line outage at $t = 5$ s; BEM vs TM. Left: Stable case, Right: Case with hyperstability. Oscillatory instability is introduced by making damping coefficient of machine negative.

Since the predictor subprocess is parallelizable, it motivates us to develop BEM-PCP and decrease the corresponding CPU-time by using parallel computing. We demonstrate that further speedup is possible using the BEM-PCP approach. Table I compares the performance of BEM-PCP with the state-of-art cascading failure simulation methods. This table claims that BEM-PCP can speedup cascade simulations while maintaining the exact accuracy as BEM-PC. The rest of the paper attempts to prove this assertion.

II. Dynamic Simulation of Cascading Failures: Preliminaries & Proposed Method

A. Preliminaries

Dynamic models of cascading failure simulation can be expressed as a set of coupled nonlinear DAEs augmented with inequalities that incorporate discrete relay actions

$$\dot{x} = f(x, V, z) \tag{1}$$

$$0 = I(x, V, z) - Y_N(z)V \tag{2}$$

$$0 \succ h(x, V, z). \tag{3}$$

Here, x and V denote the state vector of machines, and vector of real and imaginary parts of nodal voltages, respectively. The status of relays are represented by a vector of binary variables z. The real and imaginary parts of injected currents at each bus

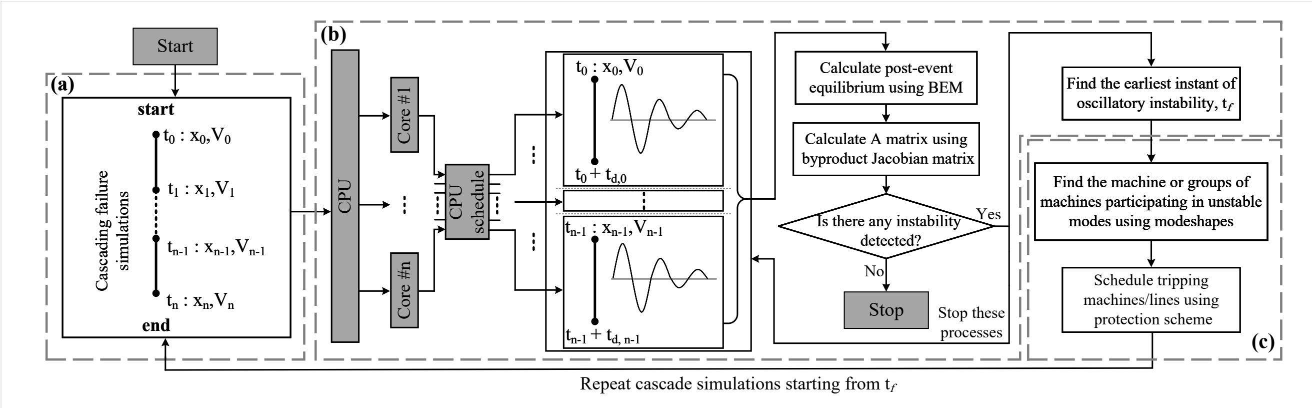

Fig. 3: Flowchart of BEM-PCP. $t = t_i, i \in \{0, 1, ..., n-1\}$: instants of tiers of cascade. (b): Parallelized predictor subprocess. (c): Corrector subprocess. Subprocess (b) is performed as parallel computation.

are indicated by vector I, and Y_N is the real form of network's admittance matrix.

In the cascading failure simulations, Initial Value Problems (IVPs) on sets of DAEs (1)-(2) are formed and solved independently following each *event*. Every discontinuity by a discrete change in the system such as relay action is called an *event*. Among two existing approaches for solving IVPs in power systems, partitioned and simultaneous [8], [9], we will rely on the simultaneous approach with implicit integration method, which can benefit efficiency of simulations using variable time-step.

B. Proposed BEM-PCP Methodology

Figure 3 shows the flowchart of BEM-PCP approach, where parallelized predictor-corrector is proposed to: 1) overcome the hyperstability issue of BEM, and 2) speedup the predictor subprocess of BEM-PC [14].

Subprocess (a) – Cascading failures simulation: In this subprocess, cascade simulations run using variable-step BEM in a serial manner. The cascading failure is triggered with initial bus outages and tripping connected lines to those buses at t_0. Following each event, IVP(s) are solved with known initial condition (x_0, V_0, z_0) for each island. Appropriate relays such as out-of-step machine protection, overcurrent (OC) relays, and undervoltage load shedding (UVLS) relays are modeled in the cascade simulations. Simulations in subprocess (a) are stopped if **i)** steady state is reached in the time-domain simulation with no expected relay action, or **ii)** the island under consideration collapses.

Subprocess (b) – Parallelized predictor: This subprocess starts after cascading simulations in subprocess (a). For each post-event island that is posed as an independent IVP, it must be verified that there is no oscillatory instability. Due to the independent nature of these IVPs, they can be solved in parallel to speedup simulations. Therefore, each IVP with corresponding known initial condition is assigned to a parallel core. Simulations are run using variable-step BEM for suitable periods $t_{d,i}$s to reach the stable or unstable equilibrium points. Relay actions that imply a discrete change in the simulations are inhibited. Once the equilibrium is reached, the system matrix (A-matrix) of the linearized model around the equilibrium is calculated using Jacobian matrix which is a by-product of BEM simulations. For more details, see [14].

For each post-event island, we inspect the eigenvalues of the A matrix for any inter-area/local oscillatory unstable mode. We are interested in the earliest instant of oscillatory instability (t_f) among all events, since results after that instant are inaccurate and have to be corrected.

Subprocess (c) – Corrector: If any oscillatory instability is detected in subprocess (b), we find out machines or group of machines that are participating in the unstable mode by investigating right eigenvectors of speeds of machines that are calculated in subprocess (b). Next, appropriate pre-specified special protection scheme (SPS) as in the ground truth is applied. Runtime of subprocess (c) in BEM-PCP is negligible.

After SPS commands are executed, we repeat simulations in subprocess (a) from t_f, and re-apply subprocesses (b) and (c) for events after t_f. The sequence of subprocesses in Fig. 3 are stopped once no oscillatory instability is detected.

Remarks on modelling:

1– *Machine model:* We consider a detailed 4^{th}-order machine model with $E_q^{'}$, $E_d^{'}$, δ, $\Delta\omega$ states integrated with static exciter and a first-order governor model with E_{fd} and P_m states [9] in BEM-PCP and all benchmarks.

2– *Relay model:* All models include identical UVLS relays for tripping loads in buses with voltages below a threshold v_{th} with delay T_{delay}^{UVLS}, OC relays for tripping overloaded lines with delay T_{delay}^{OC}, generators *out-of-step* protection, and pre-specified functional SPS actions for protections against oscillatory instability modes with trip delay T_{delay}^{SPS}.

3– *Adaptive COI-frame:* Instead of using Real-Imaginary network frame rotating at synchronous speed, in each island we project all phasors on *Center-of-Inertial (COI)*-reference frame rotating with $\omega_{coi} = \frac{1}{\sum_{i \in M} H_i} \sum_{i \in M} H_i \omega_i$ speed, where ω_i and H_i are the rotor speed and inertia constant of the ith machine. M refers to set of all machines in the island. COI-frame will add two additional states δ_{COI} and $\Delta\omega_{COI}$ to each island. For more details, such as appropriate initializations of states including speed and angle of COI-frame, please see [14]. Adaptive COI-frame is identically applied to BEM-PCP and other models.

4– Structure of cascade model: Since TM does not suffer from hyperstability, it only runs subprocess (a) in Fig. 3. However, BEM-PCP is required to run the whole process in Fig. 3 iteratively until no oscillatory instability is detected. Although BEM-PCP runs subprocesses (b), and (c) in addition to subprocess (a) that TM runs, it uses much larger step-sizes than TM due to stiff decay property, which makes BEM-PCP much faster than TM. For more details on the stiff decay property, please see [16].

III. Case Studies

The Polish system during winter $1999-2000$ peak condition [17] is used as test network to contrast results of BEM-PCP, BEM-PC [14], and the traditional approach based on TM. The Polish network has $2,383$ buses, 327 machines, and $2,896$ lines. We disconnect 3 buses and connected lines at $t = 3$ s to start cascading failure. Relay characteristics are used as in [14]. BEM-PCP is using variable step-size changing from $\Delta t_{min} = 0.02$ s to $\Delta t_{max} = 0.4$ s like BEM-PC. Monte-Carlo simulations with 500 random initial node outages are performed in AMD Ryzen 7 $3800X$ CPU with 32 GB RAM. For comparing BEM-PC against TM, 4 servers with 2.2 GHz Intel Xeon Processor, 24 CPU/server, and 128 GB RAM in PSU's ROAR facility [18] were used. In the simulations, traditional TM-based cascading failure simulation reflects the ground truth.

A. Results of Monte-Carlo Runs

As mentioned before, BEM-PCP is able to speedup BEM-PC simulations by taking advantage of parallel computation in subprocess (b) in Fig. 3, while preserving the exact accuracy as BEM-PC. Subplots in Fig. 4 demonstrate how often demand loss and number of line outages at the end of cascade are above a certain level for TM and BEM-PC/BEM-PCP. This figure exhibits almost an exact match among results of TM and BEM-PC/BEM-PCP.

Table II compares various results of BEM-PC [14] against TM for 500 Monte-Carlo simulations. Other than runtime ratios in this table, the remaining data are applicable to contrast BEM-PCP against TM (since BEM-PC and BEM-PCP produce identical results). This table presents errors in state (connected or disconnected) of buses, machines, and lines at the end of cascade. Also, it provides maximum errors in the voltage magnitude, voltage angles of buses, and frequency of machines. The central tendency measures and maximum values shows a very high accuracy for BEM-PC/BEM-PCP in replicating TM results. Path agreement measure (R) [12], [14] compares dependent branch outages (line outages after initial disturbance excluding initial disconnections) in the corresponding contingencies in TM and BEM-PC/BEM-PCP. $R = 1$ shows complete agreement in dependent line outages during cascade. Different measures for R in Table II shows that models have an almost complete agreement in cascade path. Finally, measures for runtime ratio in the table indicates that BEM-PC is on average ≈ 35 times faster than TM.

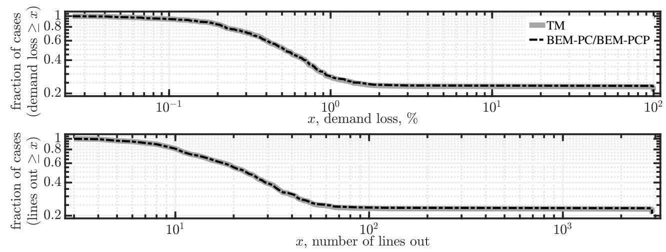

Fig. 4: Top: Fraction of cases with $\%$ demand loss $\geq x$ at the end of cascade. Bottom: Fraction of cases with line outage $\geq x$ at the end of cascade.

Fig. 5: Boxplots of runtimes of subprocesses (a) and (b) in BEM-PC as in Fig. 3 during 500 MC runs of Polish system. Runtimes are expressed as a $\%$ of total runtimes BEM-PC.

Figure 5 shows $box-whisker$ plots of runtimes of sub-processes (a), and (b) in BEM-PC expressed as a $\%$ of total runtimes of BEM-PC. It indicates that, on average, runtime of subprocess (b) is 29% of total CPU-time of each simulation for BEM-PC. This is the only part that BEM-PCP is able to speedup w.r.t. BEM-PC. Runtime of subprocess (c) in Fig. 3 is negligible. The left *box-whisker* plot in Fig. 6 represents ratios of total and subprocess (b) runtimes of corresponding cases in BEM-PC w.r.t. BEM-PCP. This plot reveals that on average BEM-PCP is $\approx$ 17% faster than BEM-PC, while achieving an average of 1.9 times speedup in subprocess (b) compared to BEM-PC. The right subplot demonstrates that for BEM-PCP in average $\approx$ 0.5%, and 3% of total runtime and CPU-time of subprocess (b) in each case can be attributed to overhead of parallel computation. When running in parallel, data need to be shared between processors and sometimes processors need to coordinate. This communication between processors increases runtime known as overhead. We used modified version of *ParTicToc* [19] in MATLAB to calculate CPU-time of individual processors in parallel computation and compute the overhead.

TABLE II: (a) End of cascade error, (b) path agreement measure (R), and (c) run time in TM w.r.t. BEM-PC [14]

		mean	min	max	median
error in state of	buses	0.1220	0	8	0
	machines	0.0620	0	3	0
	lines	0.1600	0	7	0
maximum error in	$\lvert v \rvert, pu$	0.0008	0	0.0360	$2.1e{-5}$
	$\angle v, deg.$	0.1441	0	10.1075	$4.0e{-4}$
	f, Hz	0.0165	0	0.2414	$8.4e{-5}$
R		0.9922	0.75	1	1
runtime ratio		**34.6097**	**1.1593**	**430.3984**	**24.7959**

TABLE III: End-of-cascade comparison: Oscillatory instability case. Runtimes are normalized w.r.t. runtime of BEM-PCP.

Approach	demand loss	lines out	mach. Out	cascade duration, s	runtime ratio
TM	0.92	35	3	115.66	45.94
BEM	0.18	10	0	43.69	0.18
BEM-PC	0.92	35	3	115.96	1.14
BEM-PCP	0.92	35	3	115.96	1

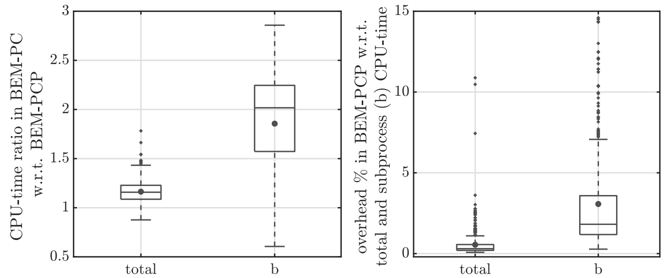

Fig. 6: Left: Boxplots of ratios of total and subprocess (b) runtimes of BEM-PC w.r.t BEM-PCP for MC runs. Right: Boxplots of runtimes of *overhead of parallel computation* in BEM-PCP expressed as a % of total and subprocess (b) runtimes of BEM-PCP for MC runs.

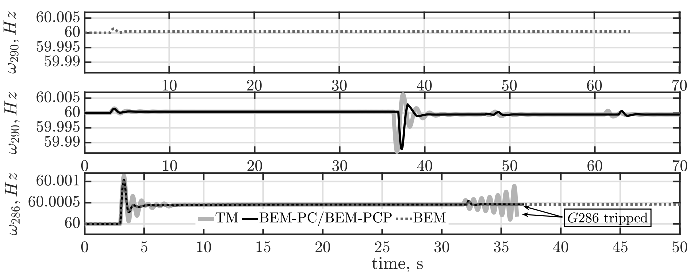

Fig. 7: Hyperstability case: Speeds of $G286$ and $G290$ in system with oscillatory instability in the middle of cascade.

B. Oscillatory Instability Challenge & Efficiency of BEM-PCP

In this section, the accuracy of BEM-PCP is tested against a case with oscillatory instability in the system. Simulations for this section were ran in PSU's ROAR facility [18]. In the tested scenario, we create oscillatory instability by making damping coefficient of some of machines negative at some time in the middle of cascade. The pre-specified SPS that trips two unstable machines with highest magnitude of oscillations is applied after 4.5 s upon detection of unstable mode. In TM this happens through an explicit SPS action. However, in the BEM-PC and BEM-PCP the functional SPS actions are performed through eigendecomposition of system matrix and finding participating machine(s) in the oscillatory unstable mode in subprocess (b). Figure 7 and Table III compare results of BEM-PCP, BEM-PC, TM, and BEM without PC approach. Figure 7 plots speeds of $G286$ and $G290$ as ω_{286} and ω_{290}. As can be seen in the figure, cascade path for BEM is different with others. However, BEM-PCP is able to follow exact path and end results of cascade as in TM. SPS in TM and BEM-PCP trips unstable machine $G286$, 4.5 s upon detection of oscillatory instability. Table III indicates that the end results of cascade for the test case with oscillatory instability is identical for TM, BEM-PC, and BEM-PCP where BEM-PCP is much faster than TM, and it is able to speedup BEM-PC by 14%.

IV. Conclusion

BEM-PC was recently proposed as an approach for speeding up dynamic simulations of cascading failures in power systems. Results of simulations for BEM-PC represented a significant speedup with high accuracy in replicating cascade simulation results of traditional methods such as TM. This paper demonstrated that a further speedup in BEM-PC is possible through a parallelized version of BEM-PC which is called BEM-PCP. The parallelized version is able to accelerate the predictor subprocess of BEM-PC using multiple parallel processors. Monte-Carlo studies on a $2,383$-bus Polish system indicated that BEM-PCP is on average 17% faster than BEM-PC while producing identical results as BEM-PC.

References

[1] P. Henneaux *et al.*, "Benchmarking quasi-steady state cascading outage analysis methodologies," *2018 IEEE Int. Conf. on Prob. Methods Applied to Power Systems (PMAPS)*, pp. 1–6, 2018.

[2] S. Gharebaghi, S. G. Vennelaganti, N. R. Chaudhuri, T. He, and T. F. L. Porta, "Inclusion of pre-existing undervoltage load shedding schemes in ac-qss cascading failure models," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5645–5656, 2021.

[3] A. J. Flueck *et al.*, "Dynamics and protection in cascading outages," in *2020 IEEE Power Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[4] Wang, Leibao, Hu, Bo, and Xie, Kaigui, "Dynamic cascading failure model for blackout risk assessment of power system with renewable energy," *E3S Web Conf.*, vol. 257, p. 01 072, 2021.

[5] C. Parmer *et al.*, "Developing a dynamic model of cascading failure for high performance computing using trilinos," *Association for Computing Machinery*, pp. 25–34, 2011.

[6] S. K. Khaitan *et al.*, "Fast parallelized algorithms for on-line extended-term dynamic cascading analysis," *2009 IEEE/PES Power Systems Conference and Exposition*, pp. 1–7, 2009.

[7] "A network-based structure-preserving dynamical model for the study of cascading failures in power grids," *Electric Power Systems Research*, vol. 209, p. 107 987, 2022, ISSN: 0378-7796.

[8] P. Kundur, *Power system stability and control*. NY, USA: McGraw-Hill, 1994.

[9] P. W. Sauer, M. A. Pai, and J. H. Chow, *Power system dynamics and stability: with synchrophasor measurement and power system toolbox*. John Wiley & Sons, 2021.

[10] M. Vallem *et al.*, "Hybrid cascading outage analysis of extreme events with optimized corrective actions," *Int. Conf. on Intelligent System App. to Power Systems (ISAP)*, pp. 1–6, 2017.

[11] P. Henneaux *et al.*, "A two-level probabilistic risk assessment of cascading outages," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2393–2403, 2016.

[12] J. Song *et al.*, "Dynamic modeling of cascading failure in power systems," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2085–2095, 2016.

[13] Y. Liu *et al.*, "Modeling and analysis of cascading failures in large-scale power grids," in *2022 IEEE Kansas Power and Energy Conference (KPEC)*, 2022, pp. 1–6.

[14] S. Gharebaghi, N. R. Chaudhuri, T. He, and T. La Porta, "An approach for fast cascading failure simulation in dynamic models of power systems," *Applied Energy*, vol. 332, p. 120 534, 2023.

[15] N. R. Chaudhuri, S. Gharebaghi, T. He, and T. F. La Porta, *Fast Contingency Simulation in Dynamic Models of Power Systems*. Provisional U. S. Patent Serial No. 63/336,810, filed on April 29, 2022.

[16] U. M. Ascher and L. R. Petzold, *Computer methods for ordinary differential equations and differential-algebraic equations*. Siam, 1998.

[17] R. D. Zimmerman *et al.*, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans.Power Syst.*, vol. 26, no. 1, pp. 12–19, 2011.

[18] *Penn State, Institute for computational and data sciences*. [Online]. Available: https://www.icds.psu.edu/computing-services/.

[19] *ParTicToc, timing PARFOR loops.* [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/27472-partictoc.