Performance assessment of shooting methods using parallel cloud computing

Agundis-Tinajero, Gibran; Gallardo, Rafael Peña; Segundo-Ramírez, Juan; Visairo-Cruz, Nancy; Guerrero, Josep M.

Published in:
COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering

DOI (link to publication from Publisher):
10.1108/COMPEL-08-2018-0307

Publication date:
2019

Document Version
Early version, also known as pre-print

Performance Assessment of Shooting Methods Using Parallel Cloud Computing

Gibran Agundis-Tinajero, Rafael Peña-Gallardo, Juan Segundo-Ramírez, Nancy Visairo-Cruz, and Josep M. Guerrero

Abstract

Purpose - The purpose of this study is to present the performance evaluation of three shooting methods typically applied to obtain the periodic steady-state of electric power systems, with the aim to check the benefits of the use of cloud computing regarding relative efficiency and computation time.

Design/methodology/approach - The mathematical formulation of the methods is presented, and their parallelization potential is explained. Two case studies are addressed, and the solution is computed with the shooting methods using multiple CPU cores through cloud computing.

Findings - The results obtained show a reduction in the computation time and increase in the relative efficiency by the application of these methods with parallel cloud computing, in the problem of obtainment of the periodic steady-state of electric power systems in an efficient way. Additionally, the characteristics of the methods, when parallel cloud computing is used, are shown and comparisons among them are presented.

Originality/value - The main advantage of employment of parallel cloud computing is a significant reduction of the computation time in the solution of the problem of a heavy computational load caused by the application of the shooting methods.

Index Terms

Cloud computing, parallel computing, periodic steady-state, shooting methods, electric power systems.

I. Introduction

THE computation of the periodic steady-state solution of electrical networks can be a difficult task. A common way to do it, is to integrate over time the differential equations, which represent the dynamic of the system, until the initial transient disappears. However, this approach, which is known as the Brute Force (BF) method (Parker and Chua, 2012), needs a large computation time (CPU time), besides the system has to be stable and the initial conditions have to be close to the system attractor of the limit cycle.

In this way, fast time-domain methods based on the Poincaré map, also known as shooting or Newton methods (Semlyen and Medina, 1995, Segundo-Ramirez and Medina, 2010a), have been developed to cope with the drawbacks of the BF method. The solution obtained by using the shooting methods is approximated as a two-point boundary-value, solving it through an iterative Newton method. These methods have the advantages that the system does not need to be stable, the initial conditions are not as sensitive as in the case of the BF method, and quadratic convergence is achieved, having small CPU times (Agundis-Tinajero et al., 2018).

Nevertheless, the advantages of using the shooting methods for the steady-state computation of electrical networks can be eclipsed for large and/or stiff systems. This is because the computational burden increases, and the computing time needed to solve the methods becomes a drawback.

In this regard, several works have been done in order to increase the efficiency of the shooting methods and to overcome the above-mentioned problems. In (Medina and Garcia, 2004) is proposed to compute the transition matrix only once, then the shooting method considers this transition matrix for the following

Gibran Agundis-Tinajero, Rafael Peña-Gallardo, Juan Segundo-Ramírez, and Nancy Visairo-Cruz are with the Universidad Autónoma de San Luis Potosí, San Luis Potosí, S.L.P., México.

Josep M. Guerrero is with the Aalborg University, Aalborg, 9200 Denmark.

Newton iterations, avoiding the computation of the transition matrix on each iteration. This simplification reduces the computation time but increases the number of iterations needed for the method to converge, and in some cases the method is unable to find a solution. In (Segundo-Ramírez and Medina, 2010b), the authors take advantage of the half-waveform symmetry of the excitation signal to halve the computation time required by the numerical differentiation method. Additionally, in (Peña et al., 2014), the LAPACK libraries are used to code efficiently the algorithms.

An alternative to reduce the computational time spent by the shooting methods is the use of parallel processing. Several contributions have been made in this sense. For example, in (Medina et al., 2003, Garcia and Acha, 2004, Medina et al., 2006), a parallel shooting method was used to obtain the periodic steady-state solution of large-scale electric systems with nonlinear components; in (Peña and Medina, 2010) a parallel shooting method was applied to obtain the steady-state solution of the induction machine after being subject to disturbances; and in (Rafael et al., n.d., Garcia and Acha, 2017) the steady-state solution of electric power systems with wind parks was achieved by applying a shooting method and parallel processing.

In order to increase the efficiency of the shooting methods, in this paper, cloud computing through the parallel computing toolbox of MATLAB (*Parallel Computing Toolbox, MATLAB*, n.d.) is used. Cloud computing can be defined as a computing environment where computing needs from a user can be outsourced by an external provider, and all the services are provided through the internet. One of the main advantages of these scheme is that users do not have to pay for infrastructure, installation, maintenance, among others, all is covered by the provider. Fig. 1 shows the scheme used for parallel cloud computing. One advantage of this proposal, apart from reducing the computation time, is the possibility to solve electric networks with a high number of components and without the limitation in the number or processors available in a single computer or computer center.

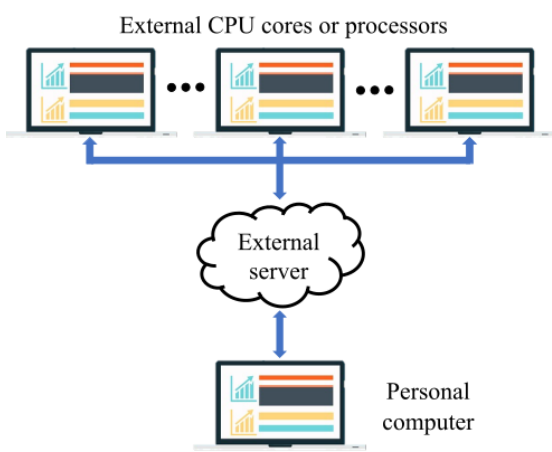

Fig. 1. Parallel cloud computing scheme.

Additionally, the performance evaluation of three shooting methods is presented. At the best knowledge of the authors and after a careful review of the state of the art, cloud computing has not been used to increment the efficiency of the shooting methods in their application to power systems.

The rest of the paper is organized as follows: Section II shows a review of the shooting methods used in this paper. Section III present an explanation of how the parallel computing formulation is done. Section IV outlines the case studies and presents the obtained results. Finally, Section V provides the conclusions of this work.

II. Shooting methods: a review

The general formulation of nonlinear time-varying systems, such as the electric power systems, can be given by a set of ordinary differential equations, of the form (Parker and Chua, 2012),

$$\dot{\mathbf{x}} = \mathbf{f}(t, \mathbf{x}) \tag{1}$$

where $\mathbf{x}$ is the state vector of n elements and t is the time.

If the system is periodic, the steady-state solution will have a fundamental period T, in such a way, that from Eq. (1), in steady-state, the following relation is satisfied (Lian and Noda, 2010),

$$\mathbf{x}_T = \mathbf{x}_0 \tag{2}$$

where $\mathbf{x}_0$ is the state vector at t_0, and $\mathbf{x}_T$=$\mathbf{x}$($t_0 + T$, t_0; $\mathbf{x}_0$).

On the other hand, the representation of dynamical periodic systems can also be performed by using a Poincaré map $\mathbf{P}(\mathbf{x})$ (Nayfeh and Balachandran, 2008). Therefore, Eq. (2) can be rewritten as follows,

$$\mathbf{x}_T = \mathbf{P}(\mathbf{x}_0) \tag{3}$$

Therefore, a way to compute the periodic state solution of Eq. (1) is based on finding an initial condition such that,

$$\mathbf{P}(\mathbf{x}_0) - \mathbf{x}_0 = 0 \tag{4}$$

In this approximation, the problem of computing the periodic steady-state solution with a period T, becomes in a two-point boundary-value problem (Garcia and Acha, 2017). This can be solved through an iterative Newton method as follows,

$$\mathbf{x}_0^{i+1} = \mathbf{x}_0^i - \left(\left. \frac{\partial \mathbf{P}(\mathbf{x}_0)}{\partial \mathbf{x}_0} \right|_{\mathbf{x}_0 = \mathbf{x}_0^i} - \mathbf{I} \right)^{-1} \left(\mathbf{P}(\mathbf{x}_0^i) - \mathbf{x}_0^i \right) \tag{5}$$

where $\mathbf{I}$ represents an identity matrix of size $n \times n$, $\mathbf{x}_0^i$ is the initial condition, and $\mathbf{P}(\mathbf{x}_0^i)$ is obtained by integrating Eq. (1) during one full cycle with the initial condition $\mathbf{x}_0^i$. When two consecutive state vectors meet a convergence criterion error, then, the periodic steady-state solution is found (Parker and Chua, 2012).

In Eq. (5), the Jacobian of the Poincaré map is also known as the state transition matrix $\mathbf{\Phi}$, and it can be defined as,

$$\mathbf{\Phi} = \left. \frac{\partial \mathbf{P}(\mathbf{x}_0)}{\partial \mathbf{x}_0} \right|_{\mathbf{x}_0 = \mathbf{x}_0^i} \tag{6}$$

which also can be approximated by finite differences as follows (Nayfeh and Balachandran, 2008),

$$\mathbf{\Phi} \approx \left. \frac{\Delta \mathbf{x}_T}{\Delta \mathbf{x}} \right|_{\mathbf{x} = \mathbf{x}_0^i} \approx \left. \frac{\Delta \mathbf{P}(\mathbf{x})}{\Delta \mathbf{x}} \right|_{\mathbf{x} = \mathbf{x}_0^i} \tag{7}$$

In order to compute $\mathbf{\Phi}$, some methods have been reported (Medina et al., 2013). For all the methods, numerical integration processes are needed (Magaña-Lemus et al., 2013). Each method presents a different approximation, requiring the computation of a different number of full cycles for the full identification of $\mathbf{\Phi}$.

A concise description of the methods used in this work to compute the transition matrix is given below.

A. Numerical differentiation method

The numerical differentiation (ND) method (Semlyen and Medina, 1995), search a numerical approximation of $\Delta\mathbf{x}_T|_{\mathbf{x}=\mathbf{x}_0^i}$ and $\Delta\mathbf{x}|_{\mathbf{x}=\mathbf{x}_0^i}$ as follows,

$$\Delta\mathbf{x}|_{\mathbf{x}=\mathbf{x}_0^i} = \mathbf{x}(t_0, t_0; \mathbf{x}_0^i + \Delta\mathbf{x}_0^i) - \mathbf{x}(t_0, t_0; \mathbf{x}_0^i) \tag{8}$$

$$\Delta\mathbf{x}_T|_{\mathbf{x}=\mathbf{x}_0^i} = \mathbf{x}(t_0 + T, t_0; \mathbf{x}_0^i + \Delta\mathbf{x}_0^i) - \mathbf{x}(t_0 + T, t_0; \mathbf{x}_0^i) \tag{9}$$

If the perturbation vector $\Delta\mathbf{x}_0^i$ is selected as εU_k, being ε a small real number and U_k the k-th column of the identity matrix of size n, where $k = 1, 2, ..., n$, then Eqs. (8) and (9) can be rewritten as follows,

$$\Delta\mathbf{x}|_{\mathbf{x}=\mathbf{x}_0^i} = \varepsilon U_k \tag{10}$$

$$\Delta\mathbf{x}_T|_{\mathbf{x}=\mathbf{x}_0^i} = \mathbf{P}(\mathbf{x}_0^i + \varepsilon U_k) - \mathbf{P}(\mathbf{x}_0^i) \tag{11}$$

In accordance with Eqs. (10) and (11), Eq. (7) becomes,

$$\mathbf{P}(\mathbf{x}_0^i + \varepsilon U_k) - \mathbf{P}(\mathbf{x}_0^i) \approx \mathbf{\Phi}\varepsilon U_k \tag{12}$$

where $\Phi_k = \mathbf{\Phi} U_k$ is the k-th column of $\mathbf{\Phi}$ and can be approximated as (Daz-Araujo et al., 2018),

$$\Phi_k \approx \frac{\mathbf{P}(\mathbf{x}_0^i + \varepsilon U_k) - \mathbf{P}(\mathbf{x}_0^i)}{\varepsilon} \quad \forall\ k = 1, 2, ..., n \tag{13}$$

Each column of $\mathbf{\Phi}$ is computed using Eq. (13). The n states of Eq. (1) have to be perturbed separately to compute the n columns of the transition matrix. Note that $n+1$ full cycles have to be computed before Eq. (5) can be applied (Semlyen and Shtash, 2000).

B. Enhanced numerical differentiation method

The enhanced numerical differentiation (END) method (Segundo-Ramírez and Medina, 2010b), takes advantage of the half-wave symmetry of the input waveforms, such as voltage and current, to improve the efficiency of the ND method. This method consist in the evaluation of Eq. (7) with the approximation of $\mathbf{x}(t_0 + T, t_0; \mathbf{x}_0^i)$ through the extrapolation of $\mathbf{x}(t_0 + T/2, t_0; \mathbf{x}_0^i)$.

Using this method, the integration of Eq. (1) for the computation of $\mathbf{\Phi}$, it is done in $T/2$ instead of T, which increases the efficiency of the ND method. In this way, the integration of Eq. (1) is needed for $(n+1)/2$ full cycles to complete the computation of the transition matrix for each Newton iteration (Cisneros-Magaña et al., 2014).

C. Discrete exponential expansion method

The discrete exponential expansion (DEE) method was proposed in (Segundo-Ramirez and Medina, 2010a). In this method the transition matrix $\mathbf{\Phi}$ is obtained following a procedure of identification step-by-step based in a recursive formulation. The identification of $\mathbf{\Phi}$ requires the integration of only one full cycle of period T. In this method the approximation is carried-out as follows (Segundo-Ramirez and Medina, 2009b),

$$\mathbf{\Phi} \approx \prod_{m=1}^{K} e^{\mathbf{F}_m \Delta t_m} \tag{14}$$

the Jacobian matrix $\mathbf{F}_m$ is,

$$\mathbf{F}_m = \mathbf{F}(t, x(t))|_{t=\frac{t_m + t_{m-1}}{2}, x=\frac{x(t_m)+x(t_{m-1})}{2}} \tag{15}$$

where Δt_m is defined as $t_m - t_{m-1}$, K is the number of intervals in a period T, and t_m is the m-th element of the time vector from t_0 to $t_0 + T$.

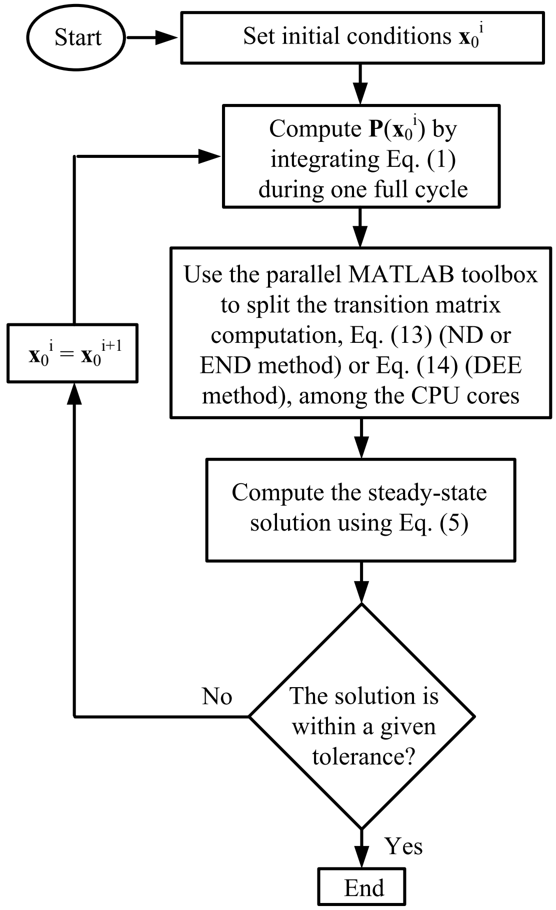

Fig. 2. Flowchart of the shooting methods with parallel computing.

III. Parallel cloud computing

Shooting methods require the construction of the transition matrix, which is one of the most time-consuming tasks, since it depends on the number of the system equations and it relies on numerical integration processes. However, the transition matrix can be obtained using parallel processing, reducing in this way the computing time. In this approach, the transition matrix is approximated by creating individual tasks, i.e., each individual task can be computed separately and does not need information from another task.

As can be seen from Eq. (13), in the ND and END methods, the approximation of the transition matrix is computed column by column, and the information needed to compute each column is independent from the others. Besides, in the DEE method, Eq. (14) shows that the transition matrix is computed by means of the multiplication of matrices, but each matrix is independent from each other. These characteristics bring the opportunity to take advantage of ability to parallelize these numerical methods, in particular, for the computation of the transition matrix.

In this regard, the parallel computing toolbox of MATLAB can be used to parallelize the shooting method algorithms by splitting the transition matrix computation among CPU cores or processors. To perform the parallelization, the *parfor* command of MATLAB is used, this command splits the execution of *for-loop* iterations over the CPU cores, also known as "workers", in a parallel pool. Additionally, an important requirement for using *parfor-loops* is that individual iterations must be independent.

On the other hand, cloud computing services can be used together with the parallel computing toolbox of MATLAB. Cloud servers provide CPU cores required by the user and performs the simulation in the same MATLAB environment. This advantage eases the use of these advanced computing tools, since the user do not need to have physically the hardware required for the parallel implementation, i.e. multiple

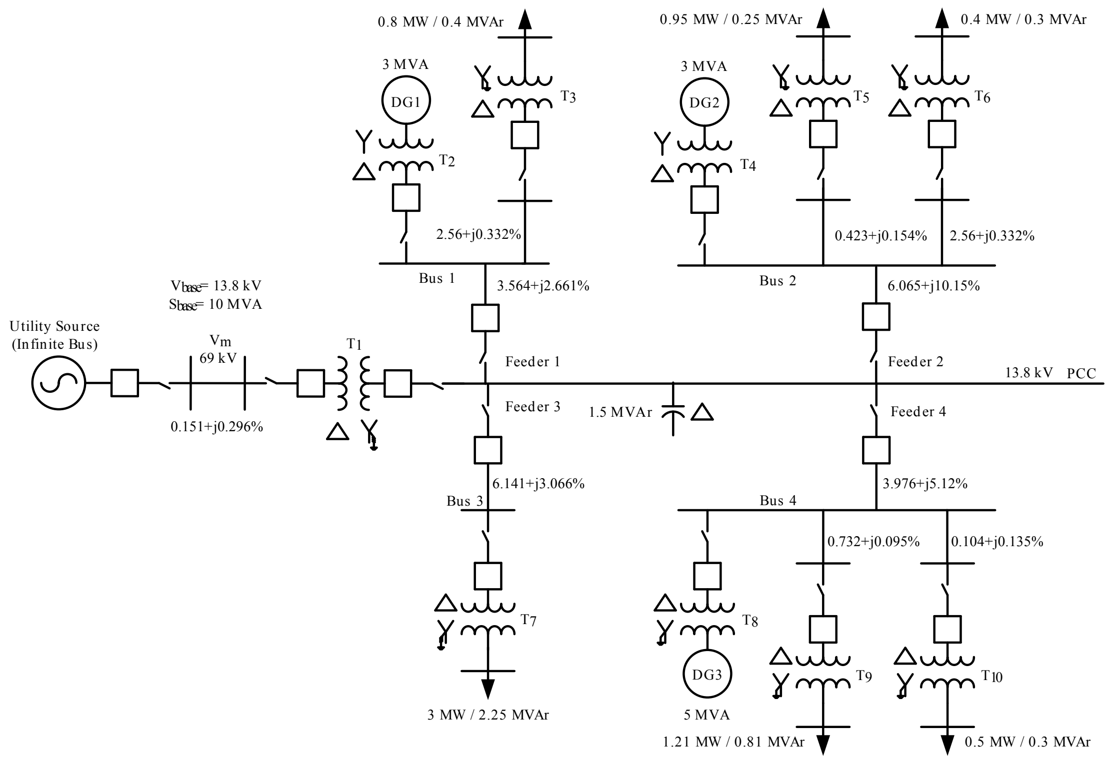

Fig. 3. Single line diagram of the test system (Katiraei and Iravani, 2006).

CPU cores, instead the user can use external CPU cores and run the simulations in the same MATLAB environment (Armbrust et al., 2009).

Fig. 2 shows the flowchart used to parallelize the shooting methods using the parallel computing toolbox of MATLAB and cloud computing.

IV. Shooting methods performance assessment

In this Section two test systems are implemented to assess the CPU time efficiency of three fast time-domain methods using parallel cloud computing. The convergence error tolerance used in the case studies is 1×10^{-8}.

The parallel cloud computing was done through SSH connection with an external computer located in the Aalborg University. The specs of the external computer are: 2 Intel(R) Xeon(R) CPU E5-2690 v3 at 2.60GHz, 384 GB DDR4 memory RAM at 2133MHz, and 24 CPU cores.

A. Case I: controlled microgrid system

Fig. 3 shows the single line diagram of the microgrid used as test system. The system is composed of three distributed generation (DG) units, each one includes a VSC (Segundo-Ramirez and Medina, 2009a) with a modulation index (m_f) of 21, for interconnecting a primary source with the microgrid. The VSC controls used are based on a droop-characteristic scheme with frequency restoration (Katiraei and Iravani, 2006). This test system is modeled with 86 ordinary differential equations. The system configuration and parameters can be consulted in (*IEEE Recommended Practice for Industrial and Commercial Power System Analysis*, 1990, Katiraei and Iravani, 2006).

In this first case study, the microgrid is operating in steady-state, then a perturbation is applied in the system state variables, varying their values by 5%, then, the computation of the periodic steady-state solution is carried out using the shooting methods. The computation of the steady-state is obtained several times, varying the number of CPU cores used to compare their relative efficiencies.

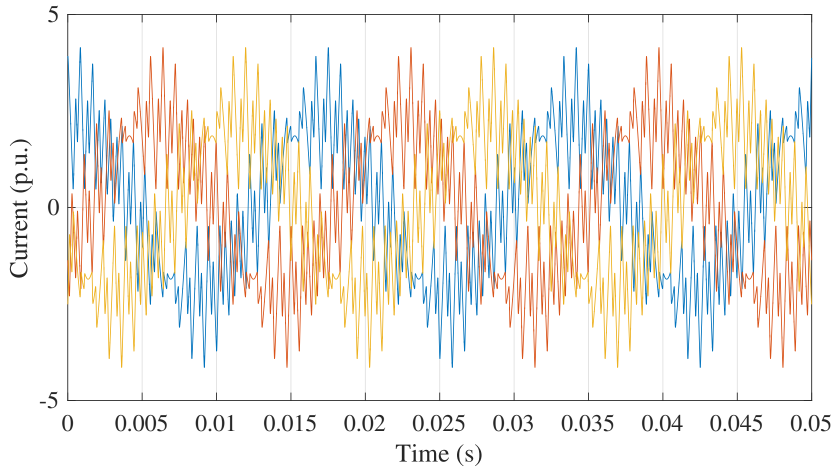

Fig. 4. DG unit 1 output currents of the controlled microgrid system.

In Table I the convergence errors of the shooting methods for this case study are shown. Note that only two iterations are needed to achieve the steady-state solution, with an error in the order of 1×10^{-9}. Furthermore, Fig. 4 shows three full cycles of the DG unit 1 output currents after the application of the Newton methods, where it can be noticed that the periodic steady-state is reached even in presence of harmonic distortion.

TABLE I
CONVERGENCE ERRORS OF CASE I

Iteration	DEE	ND	END
1	1.73×10^{-4}	1.72×10^{-4}	8.32×10^{-5}
2	7.81×10^{-9}	7.43×10^{-9}	5.98×10^{-9}

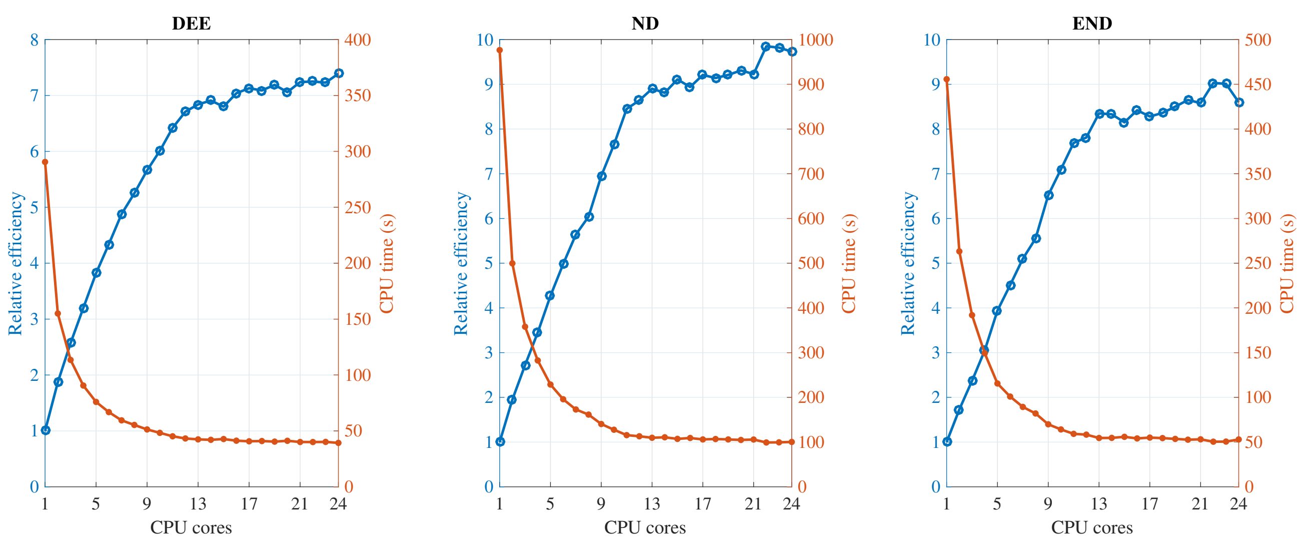

Fig. 5. Relative efficiency and CPU time required by the shooting methods in Case I.

Fig. 5 shows the relative efficiency and CPU time required by the shooting methods in the computation of the steady-state solution, as the number of CPU cores increases. It is observed that the relative efficiency for the three methods increases up approximately to the 12th core, then the relative efficiency behavior flattens with small improvements until the 24th core. The best performance regarding the relative efficiency is obtained with the ND method, with an improvement of almost 9.73 times in its efficiency, followed closely by the END method with a relative efficiency of 8.6, and finally, the DEE method improvement is only of 7.39 times its relative efficiency, having the worst performance. In regard to the CPU time, the DEE method requires only 39.28 seconds to obtain the solution, while the ND and END methods require 100.3 seconds and 52.94 seconds, respectively, using 24 CPU cores.

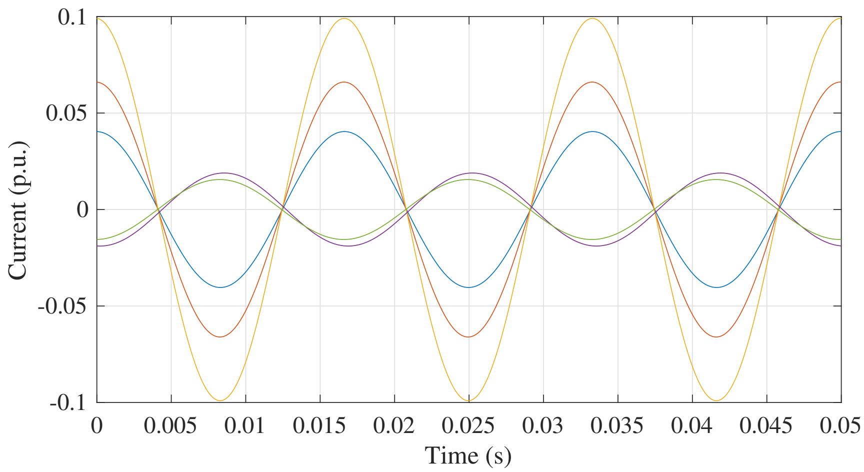

Fig. 6. Steady-state currents at the transmission lines of the IEEE 118-bus benchmark.

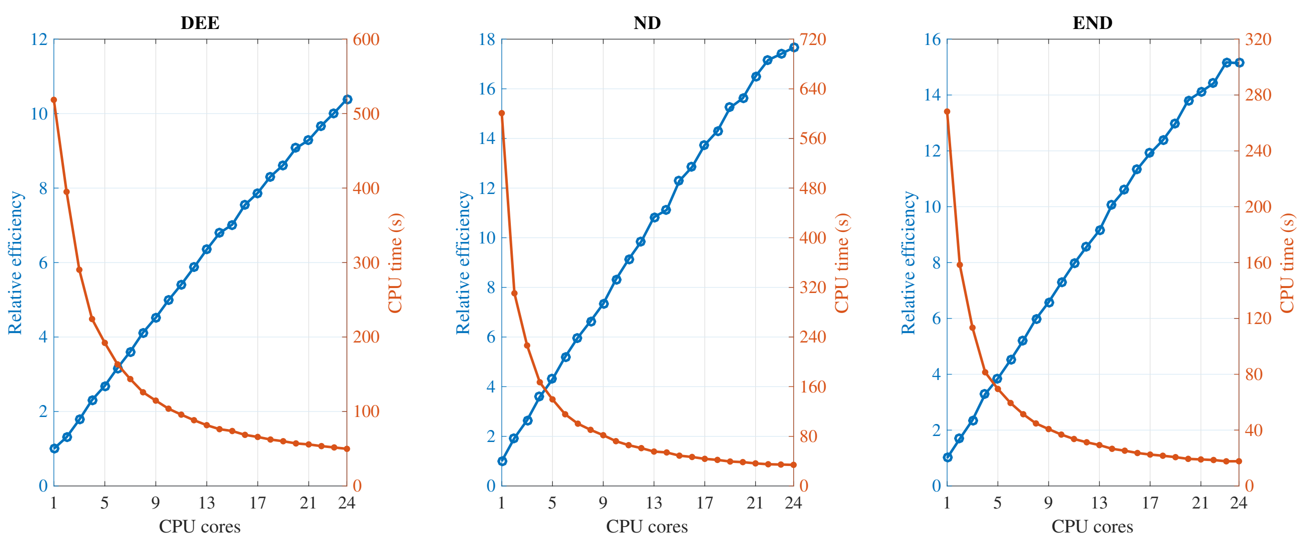

Fig. 7. Relative efficiency and CPU time required by the shooting methods in Case II.

B. Case II: IEEE 118-bus test system

In this case the IEEE test system of 118 nodes is used, it consists in 186 transmission lines, 118 capacitor banks, 54 generators and 9 magnetizing branches. The system is simulated from its start-up, and one full-cycle after this, it is computed the periodic steady-state solution with the shooting methods. This test system is modeled with 355 ordinary differential equations.

Table II shows the convergence errors of the methods in this case study. It is observed that the DEE method requires 5 iterations to achieve the steady-state with an error of 1.66×10^{-9}, while the ND and END methods require 2 iterations with errors of 2.17×10^{-14} and 1.47×10^{-14}, respectively. Fig. 6 shows three steady-state full cycles of the current at different transmission lines of the test system.

TABLE II
CONVERGENCE ERRORS OF THE CASE II

Iteration	DEE	ND	END
1	1.73×10^{-3}	1.44×10^{-8}	1.25×10^{-8}
2	4.00×10^{-5}	2.17×10^{-14}	1.47×10^{-14}
3	1.26×10^{-6}	-	-
4	4.44×10^{-8}	-	-
5	1.66×10^{-9}	-	-

The results obtained in this case study in terms of relative efficiency and CPU time required by the shooting methods are shown in Fig. 7. From this figure, it can be seen that due to the size of the test system, even with the 24 CPU cores the relative efficiency is increasing. However, it is also observed that the CPU time graph is getting flat, therefore, despite that the methods achieve better relative efficiencies, the CPU time needed will be similar using more CPU cores. This behavior is due the communication time between cores and other aspects associated with the parallelization of the methods.

In this case the DEE, ND and END methods achieve relative efficiencies of 10.39, 17.66 and 15.14, respectively. Where the ND method has the best relative efficiency and the DEE method the worst. Regarding the CPU time, the DEE, ND and END methods achieve the steady-state solution in 49.86 seconds, 34.02 seconds and 17.71 seconds, respectively.

The results obtained in both case studies show the advantages of using parallel processing through cloud computing. Besides, the shooting methods are adequate to be parallelized. In the case of this research, with 24 CPU cores, in both cases the best relative efficiency obtained was 17.66 (Case II), and the worst 7.39 (Case I). Additionally, it is observed that in Case I, despite that the DEE method has the lower increase in the relative efficiency, it was the quickest method; however, for the Case II, it needed more CPU time than the ND and END methods. Therefore, this finding indicates that for large systems the ND and END methods are more efficient.

V. CONCLUSION

In this paper the performance assessment of three shooting methods for the steady-state solution computation using parallel cloud computing is presented. It is shown, through case studies the capacity of MATLAB to parallelize processes. Furthermore, the cloud computing feature allows to extract all the parallel computing potential without the need of specialized computing equipment by the user.

The results of the case studies show the improvements achieved in terms of relative efficiency and CPU time when parallel cloud computing is applied to the shooting methods algorithms. In the first case, the best relative efficiency obtained was 9.73 using the ND method, while the worst was 7.39 from the DEE method. However, regarding the CPU time, the DEE method required less computing time in comparison with the other methods. On the other hand, the best relative efficiency in Case II was 17.66 from the ND method, and the worst was 10.39 from the DEE method. Regarding the CPU time, the END method required only 17.71 seconds, being the quickest method.

Therefore, the use of cloud computing helps to cope some of the disadvantages of the shooting methods, making them an excellent tool for the periodic steady-state solution computation even for large and/or stiff electric power systems and in the presence of harmonic distortion.

References

Agundis-Tinajero, G., Segundo-Ramrez, J., Pea-Gallardo, R., Visairo-Cruz, N., Nez-Gutirrez, C., Guerrero, J. M. and Savaghebi, M. (2018). Harmonic issues assessment on pwm vsc-based controlled microgrids using newton methods, *IEEE Transactions on Smart Grid* **9**(2): 1002–1011.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I. and Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing, *Tech. Rep. UCB/EECS-2009-28, EECS Department, U.C. Berkeley* pp. 1–23.

Cisneros-Magaña, R., Medina, A. and Segundo-Ramirez, J. (2014). Efficient time domain power quality state estimation using the enhanced numerical differentiation newton type method, *International Journal of Electrical Power & Energy Systems* **63**: 414 – 422.

Daz-Araujo, M., Medina, A., Cisneros-Magaa, R. and Ramrez, A. (2018). Periodic steady state assessment of microgrids with photovoltaic generation using limit cycle extrapolation and cubic splines, *Energies* **11**(8).

Garcia, N. and Acha, E. (2004). Periodic steady-state analysis of large-scale electric systems using Poincaré map and parallel processing, *IEEE Transactions on Power Systems* **19**(4): 1784–1793.

Garcia, N. and Acha, E. (2017). On the efficient calculation of the periodic steady-state response of grid-connected wind parks, part i, *IEEE Transactions on Sustainable Energy* **8**(2): 458–467.

IEEE Recommended Practice for Industrial and Commercial Power System Analysis (1990). *IEEE Std 399-1990* pp. 1–384.

Katiraei, F. and Iravani, M. (2006). Power management strategies for a microgrid with multiple distributed generation units, *IEEE Transactions on Power Systems* **21**(4): 1821–1831.

Lian, K. L. and Noda, T. (2010). A time-domain harmonic power-flow algorithm for obtaining nonsinusoidal steady-state solutions, *IEEE Transactions on Power Delivery* **25**(3): 1888–1898.

Magaña-Lemus, E., Medina-Ros, A., Ramos-Paz, A. and Montesinos-Gonzlez, V. H. (2013). Periodic steady state determination of power systems using graphics processing units, *2013 10th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pp. 274–279.

Medina, A. and Garcia, N. (2004). Fast time domain computation of the periodic steady-state of systems with nonlinear and time-varying components, *International Journal of Electrical Power and Energy Systems* **26**(8): 637 – 643.

Medina, A., Ramos-Paz, A. and Fuerte-Esquivel, C. (2003). Periodic steady state solution of electric systems with nonlinear components using parallel processing, *IEEE Transactions on Power Systems* **18**(2): 963–965.

Medina, A., Ramos-Paz, A. and Ruben Fuerte-Esquivel, C. (2006). Efficient computation of the periodic steady state solution of nonlinear electric systems applying parallel processing techniques, *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering* **25**(4): 900–915.

Medina, A., Segundo, J., Ribeiro, P., Xu, W., Lian, K., Chang, G., Dinavahi, V. and Watson, N. (2013). Harmonic analysis in frequency and time domain, *IEEE Transactions on Power Delivery* **28**(3): 1813–1821.

Nayfeh, A. and Balachandran, B. (2008). *Applied Nonlinear Dynamics: Analytical, Computational and Experimental Methods*, Wiley Series in Nonlinear Science, Wiley.

Parallel Computing Toolbox, MATLAB (n.d.).
URL: *https://la.mathworks.com/products/parallel-computing.html*

Parker, T. and Chua, L. (2012). *Practical Numerical Algorithms for Chaotic Systems*, Springer New York.

Peña, R. and Medina, A. (2010). Fast steady-state parallel solution of the induction machine, *Electrical Machines (ICEM), 2010 XIX International Conference on*, IEEE, pp. 1–5.

Peña, R., Núñez, J. and Medina, A. (2014). Using a newton method and LAPACK libraries to initialize electromagnetic transient simulations in power systems, *Simulation Modelling Practice and Theory* **42**: 12–18.

Rafael, P., Aurelio, M. and Olimpo, A. L. (n.d.). A methodology for the efficient computer representation of dynamic power systems: Application to wind parks, *Wind Energy* **16**(1): 109–121.

Segundo-Ramirez, J. and Medina, A. (2009a). Modeling of FACTS devices based on SPWM VSCs, *IEEE Transactions on Power Delivery* **24**(4): 1815–1823.

Segundo-Ramirez, J. and Medina, A. (2009b). Periodic steady state solution of facts devices based on spwm vscs, *2009 IEEE International Symposium on Industrial Electronics*, pp. 1644–1649.

Segundo-Ramirez, J. and Medina, A. (2010a). Computation of the steady state solution of nonlinear power systems by extrapolation to the limit cycle using a discrete exponential expansion method, *International Journal of Nonlinear Sciences and Numerical Simulation* **11**(8): 655–660.

Segundo-Ramírez, J. and Medina, A. (2010b). An enhanced process for the fast periodic steady state solution of nonlinear systems by Poincaré map and extrapolation to the limit cycle, *International Journal of Nonlinear Sciences and Numerical Simulation* **11**(8): 661–670.

Semlyen, A. and Medina, A. (1995). Computation of the periodic steady state in systems with nonlinear components using a hybrid time and frequency domain methodology, *IEEE Transactions on Power Systems* **10**(3): 1498–1504.

Semlyen, A. and Shtash, M. (2000). Principles of modular harmonic power flow methodology, *IEE Proceedings - Generation, Transmission and Distribution* **147**(1): 1–6.