IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 16, 2020, accepted March 2, 2020, date of publication March 10, 2020, date of current version March 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979775

Complex-Array-Operation Newton Solver for Power Grids Simulations

PAOLO MAFFEZZONI, (Senior Member, IEEE), AND GIAMBATTISTA GRUOSSO, (Senior Member, IEEE)
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milan, Italy

Corresponding author: Giambattista Gruosso (giambattista.gruosso@polimi.it)

ABSTRACT This paper presents a robust and efficient technique for performing repeated power flow simulations of power networks. The method relies on a vector-based formulation of the power balance equations combined with a complex-array operation Newton solver. It is shown how the method is suitable for advanced simulations of power grids, such as probabilistic analyses, where a large number of scenarios have to be explored in reasonable simulation times. Applications to benchmark single phase networks as well as to unbalanced three phase grids are provided.

INDEX TERMS Array-operation, Newton solver, power flow, probabilistic analysis, power grid simulation, simulation.

I. INTRODUCTION

Due to the rapid evolution of power transmission/distribution networks within the context of smart grids there is today a renewed interest in developing numerically efficient techniques for power flow simulations. Such an interest is motivated in part by some peculiar features shared by smart power grids, such as the high penetration of renewable sources [1], the deployment of many smart meters and network reconfiguration possibility [2], [3], the support of new types of services (e.g. the integration of electrical vehicles) as well as the envisaged active role of consumers as power demand/provide actors [4]. A reliable and robust approach to smart grids design should account for the highly uncertain nature and variability of loads and power sources that can be described only statistically [5]–[7]. This implies repeating a huge amount of power flow analyses while considering the many potential scenarios and network configurations in reasonable simulation times. Other elements of novelty are connected with the evolution of the grid from traditional topologies to more general ones and the presence of distributed power generation. Some power flow techniques which rely on iterative relaxation solvers are tailored to specific network topologies (e.g. perfectly radial topology) and can exhibit poor convergence or even diverge when applied to general meshed topologies and or in the presence of many voltage-constraining generators [8]–[11]. For such reasons, in this paper we will investigate the power flow problem in connection with the robust and generally applicable Newton-Raphson (NR) solver [12], [13].

A promising approach to satisfy the growing demand for simulation efficiency is that of taking advantage of the current tendency to integrate multi-core processors and enable parallel computing. This can be achieved by exploiting some high-performance computing languages that support array programming, i.e. the application of operations to an entire set of values stored in arrays. Known languages allowing array operation are Fortran 90, Matlab, Perl Data Language, and Python. Most of these programs support operations directly on complex numbers.

In the Sec. II of this paper, we review the most common ways to formulate the power flow problem in connection with NR solver. Due to the non-analyticity of the complex conjugate operator (appearing in the complex power expression), the common practice is that of splitting complex power balance at node terminals into its real and imaginary parts. In this way, the calculations of partial derivatives forming the Jacobian matrix are carried out in the field of real numbers via non-elementary operations. The relatively complicated expressions of Jacobian elements along with their being developed in the field of real numbers do not allow direct implementation within software languages supporting complex-array-operations. In order to overcome such a limitation, in this paper we present a novel simulation method which is applicable to both single phase networks and three phase unbalanced grids as described in Sec. III. The method,

The associate editor coordinating the review of this manuscript and approving it for publication was Zhouyang Ren.

which is outlined in Sec. IV, relies on a vector-based representations of the power flow equations and on an original way to implement complex numbers multiplication. Thanks to these premises, we are able to derive an efficient way for calculating the elements of the Jacobian matrix via elementary operations completely developed in the complex numbers field. Unprecedented numerical efficiency can be achieved by implementing Jacobian elements calculations via array operations extended to a large set of grid nodes. The novel method, which is referred to as Complex-Array-operation Newton (CAN) solver, is suitable for those advanced simulations requiring a large number of repeated power flow analyses, as described in Sec. V. Finally, in Sec. VI, the efficiency and robustness of the CAN solver is illustrated by exploring its applications to the power flow analysis of single phase networks as well as to probabilistic simulations of unbalanced three phase grids.

II. POWER FLOW METHODS BASED ON NR SOLVER

For a power network composed of N nodes, the steady-state power flow problem consists in calculating the node voltage (complex) values producing the wanted power flow at the network terminals. Mathematically, the problem is commonly formulated by a set of nonlinear equations of the type:

$$\mathbf{f}_k(\vec{\mathbf{V}}) = \mathbf{V}_k^* \mathbf{I}_k - \mathbf{S}_k^* = \mathbf{0} \tag{1}$$

for $k = 1, \ldots, N$, where $\mathbf{V}_k$, $\mathbf{I}_k$ are voltage and current phasors at node k, vector $\vec{\mathbf{V}} = [\mathbf{V}_1, \ldots, \mathbf{V}_N]$ collects all voltages, while $\mathbf{S}_k = P_k + jQ_k$ denotes complex power at node k with P_k and Q_k being the active and reactive power, respectively. In addition to that, node currents are related to node voltages by means of

$$\mathbf{I}_k = \sum_{n=1}^{N} \mathbf{Y}_{kn}\, \mathbf{V}_n \tag{2}$$

where $\mathbf{Y}_{kn}$ are the entries of the node admittance matrix and N the number of nodes of the network.

Solution of (1) and (2) can be achieved either with iterative methods (e.g. Jacobi, Gauss-Seidel) or with Newton-Raphson (NR) based direct method. In what follows, we focus on NR method due to its robustness and applicability to any grid topology.

In practice, there are two main approaches to power flow formulation that differ for the way node voltages are represented. In the first formulation, voltages are described in polar coordinates $\mathbf{V}_k = |V_k|e^{j\delta_k}$ by their module $|V_k|$ and phase δ_k. In this case, by denoting with $Y_{kn} = |Y_{kn}|e^{j\theta_{kn}}$ the admittance elements, power flow equations (1) and (2) are transformed into the following set of $2 \times N$ real equations

$$\begin{aligned}
&|V_k|\sum_{n=1}^{N} |Y_{kn}||V_n| \cos(\delta_k - \delta_n - \theta_{kn}) - P_k = 0\\
&|V_k|\sum_{n=1}^{N} |Y_{kn}||V_n| \sin(\delta_k - \delta_n - \theta_{kn}) - Q_k = 0
\end{aligned} \tag{3}$$

The application of NR solver to polar coordinates problem (3) can exhibit some limitations. A first issue is connected with a potential lack of efficiency due to the repeated calls of implicit trigonometric functions. To cope with this issue, in some cases further mathematical developments are implemented that express partial derivatives of problem (3) in cartesian form rather than polar one [14]. Second, and more importantly, it has been shown that NR solver applied to polar coordinates (3) can exhibit convergence troubles or it can converge to non-physical ill-conditioned solutions that contain low voltages values [8]. For such reasons, polar formulation is commonly limited to the power flow analysis of single-phase circuit representing balanced transmission networks.

In the second approach, voltages are described in cartesian coordinates $\mathbf{V}_k = V_k^R + jV_k^I$ by their real and imaginary parts. In this case, by denoting with $\mathbf{Y}_{kn} = G_{kn} + jB_{kn}$ the node admittance elements, power flow equations (1) and (2) are transformed into:

$$\begin{aligned}
&V_k^R \sum_{n=1}^{N} (G_{kn}V_n^R - B_{kn}V_n^I)\\
&\quad + V_k^I \sum_{n=1}^{N} (B_{kn}V_n^R + G_{kn}V_n^I) - P_k = 0\\
&-V_k^I \sum_{n=1}^{N} (G_{kn}V_n^R - B_{kn}V_n^I)\\
&\quad + V_k^R \sum_{n=1}^{N} (B_{kn}V_n^R + G_{kn}V_n^I) + Q_k = 0.
\end{aligned} \tag{4}$$

The solution of cartesian formulation (4) by means of NR method exhibits a certain degree of robustness and for this reason it is frequently adopted in the power flow analysis of three-phase circuits modeling unbalanced transmission grids. However, the relatively complicated expressions of partial derivatives of (4) versus V_k^R and V_k^I, as needed for Jacobian matrix formation, along with their being developed in the real number field do not allow the exploitation of the complex-array-operation capabilities supported by some advanced software languages.

III. POWER GRIDS DESCRIPTION

The simulation method that we investigate in this paper is intended for the power flow analysis of both single-phase networks modeling power transmission systems and three-phase networks modeling unbalanced distribution grids. For a power grid made of N_b bar-buses, the number of nodes, and related voltages, are $N = N_b$ in the case of single-phase networks and $N = 3 \times N_b$ for three-phase nets. Node-interconnection elements, such as grid lines, transformers and phase shifters, are represented by means of their primitive admittance matrix. As an example, Fig. 1 shows the lumped π circuit commonly adopted for modeling single-phase grid lines.

The model is determined by the series impedance $\mathbf{Z}_s$ and shunt susceptance B. The primitive admittance matrix

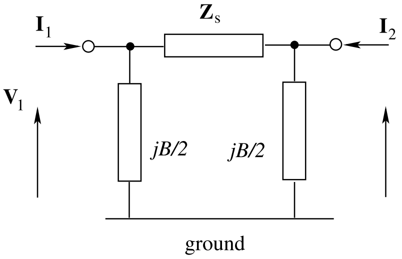

FIGURE 1. The π model of a single phase line.

associated to the circuit in Fig. 1 is the 2×2 complex matrix relating voltages to currents as follows:

$$\begin{bmatrix} \mathbf{I}_1 \\ \mathbf{I}_2 \end{bmatrix} = \begin{bmatrix} 1/\mathbf{Z}_s + jB/2 & -1/\mathbf{Z}_s \\ -1/\mathbf{Z}_s & 1/\mathbf{Z}_s + jB/2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \end{bmatrix} \tag{5}$$

The π model can be extended to three-phase lines as shown in Fig. 2. For notation compactness, the phase voltages and currents are collected into the vectors $\vec{\mathbf{V}}_1 = [\mathbf{V}_1^a, \mathbf{V}_1^b, \mathbf{V}_1^c]^T$, $\vec{\mathbf{V}}_2 = [\mathbf{V}_2^a, \mathbf{V}_2^b, \mathbf{V}_2^c]^T$ and $\vec{\mathbf{I}}_1 = [\mathbf{I}_1^a, \mathbf{V}_1^b, \mathbf{I}_1^c]^T$, $\vec{\mathbf{I}}_2 = [\mathbf{I}_2^a, \mathbf{I}_2^b, \mathbf{I}_2^c]^T$, respectively. For the three-phase line, the 6×6 primitive admittance matrix is such that:

$$\begin{bmatrix} \vec{\mathbf{I}}_1 \\ \vec{\mathbf{I}}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_{abc}^{-1} + jB_{abc}/2 & -\mathbf{Z}_{abc}^{-1} \\ -\mathbf{Z}_{abc}^{-1} & \mathbf{Z}_{abc}^{-1} + jB_{abc}/2 \end{bmatrix} \begin{bmatrix} \vec{\mathbf{V}}_1 \\ \vec{\mathbf{V}}_2 \end{bmatrix} \tag{6}$$

where $\mathbf{Z}_{abc}$ now denotes the 3×3 symmetric line impedance matrix:

$$\mathbf{Z}_{abc} = \begin{bmatrix} \mathbf{Z}_a & \mathbf{Z}_{ab} & \mathbf{Z}_{ac} \\ \mathbf{Z}_{ab} & \mathbf{Z}_b & \mathbf{Z}_{bc} \\ \mathbf{Z}_{ac} & \mathbf{Z}_{bc} & \mathbf{Z}_c \end{bmatrix}. \tag{7}$$

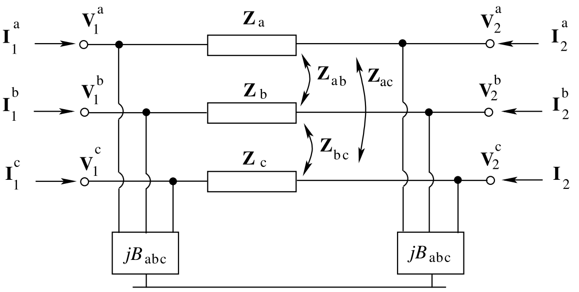

FIGURE 2. The three-phase π lines model.

Impedance matrix accounts for the series impedance of each phase line and for mutual coupling among them. Similarly, B_{abc} now denotes the 3×3 shunt susceptance matrix.

Power specification at the terminal nodes are given by loads and generators. Commonly, power flow simulation supports two types of nodes referred to as PQ and PV. In PQ nodes, power flow equations are formulated as in (1) and (2) with the active and reactive power values P_k, Q_k given as an input. Such values are determined by all of the loads connected to the node k. PV nodes, instead, are commonly used to model power generators. In PV nodes, the values of active power P_k and voltage module $|V_k|$ are given as an input. For PV nodes power flow equation (1) should be modified as we will show in the next Section.

IV. POWER FLOW BASED ON CAN SOLVER

The simulation method that we propose relies on cartesian coordinates representation of electrical quantities and power balance equations that are solved via Newton-Raphson method implemented via complex-array-operations. In order to derive the method, we start by adopting a vector representation of the complex quantities. More specifically, we use the symbol vect$(\cdot)$ to denote the operator that for any complex number or function $\mathbf{A} \in \mathbb{C}$, given as an input, provides the 2×1 output vector

$$\bar{A} = \text{vect}(\mathbf{A}) = \begin{bmatrix} A^R \\ A^I \end{bmatrix} \tag{8}$$

collecting its real and imaginary parts $A^R = \text{Re}\{\mathbf{A}\}$ and $A^I = \text{Im}\{\mathbf{A}\}$, respectively. Furthermore, we represent the multiplication of complex numbers by means of a matrix-vector operation. Given the complex numbers $\mathbf{A} = A^R + jA^I$, $\mathbf{B} = B^R + jB^I$, the vector form $\bar{C}$ of the product $\mathbf{C} = \mathbf{A}\,\mathbf{B}$ results:

$$\bar{C} = \text{vect}(\mathbf{C}) = \begin{bmatrix} C^R \\ C^I \end{bmatrix} = \begin{bmatrix} A^R & -A^I \\ A^I & A^R \end{bmatrix} \cdot \begin{bmatrix} B^R \\ B^I \end{bmatrix} \tag{9}$$

In view of (9), we now reformulate the power flow equations for PQ and PV nodes.

A. PQ NODES

For PQ nodes the power flow equations (1), (2) are transformed into the following vector form

$$\bar{f}_k = \begin{bmatrix} V_k^R & V_k^I \\ -V_k^I & V_k^R \end{bmatrix} \cdot \begin{bmatrix} I_k^R \\ I_K^I \end{bmatrix} + \begin{bmatrix} -P_k \\ Q_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{10}$$

and

$$\begin{bmatrix} I_k^R \\ I_k^I \end{bmatrix} = \sum_{n=1}^{N} \begin{bmatrix} G_{kn} & -B_{kn} \\ B_{kn} & G_{kn}^R \end{bmatrix} \cdot \begin{bmatrix} V_n^R \\ V_n^I \end{bmatrix}, \tag{11}$$

respectively.

In order to apply NR solver, the partial derivatives of (10) and (11) versus V_k^R and V_k^I should be calculated. These derivatives form the Jacobian matrix. We first concentrate on the off-diagonal terms of Jacobian. The partial derivatives of vector function $\bar{f}_k$ (i.e. imposing power balance at node kth) with respect to the real and imaginary parts of voltage at node $n \neq k$ result:

$$\frac{\partial \bar{f}_k}{\partial V_n^R} = \begin{bmatrix} V_k^R & V_k^I \\ -V_k^I & V_k^R \end{bmatrix} \cdot \begin{bmatrix} G_{kn} \\ B_{kn} \end{bmatrix} \tag{12}$$

and

$$\frac{\partial \bar{f}_k}{\partial V_n^I} = \begin{bmatrix} V_k^R & V_k^I \\ -V_k^I & V_k^R \end{bmatrix} \cdot \begin{bmatrix} -B_{kn} \\ G_{kn} \end{bmatrix}, \tag{13}$$

respectively.

Using (9), it is easily recognized that:

$$\frac{\partial \bar{f}_k}{\partial V_n^R} = \mathrm{vect}(\mathbf{V}_k^* \, \mathbf{Y}_{kn}) \tag{14}$$

and

$$\frac{\partial \bar{f}_k}{\partial V_n^I} = \mathrm{vect}(\mathbf{V}_k^* \, j\mathbf{Y}_{kn}). \tag{15}$$

This means that the off-diagonal elements of the Jacobian matrix can be calculated via multiplications of voltage phasors and admittances in the complex domain followed by some vector elements reordering. A similar result holds also for the diagonal terms of the Jacobian matrix. In fact, the partial derivatives of vector function $\bar{f}_k$, with respect to the real and imaginary parts of the voltage at the same node are:

$$\frac{\partial \bar{f}_k}{\partial V_k^R} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} I_k^R \\ I_k^I \end{bmatrix} + \begin{bmatrix} V_k^R & V_k^I \\ -V_k^I & V_k^R \end{bmatrix} \cdot \begin{bmatrix} G_{kn} \\ B_{kn} \end{bmatrix} \tag{16}$$

and

$$\frac{\partial \bar{f}_k}{\partial V_k^I} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} I_k^R \\ I_k^I \end{bmatrix} + \begin{bmatrix} V_k^R & V_k^I \\ -V_k^I & V_k^R \end{bmatrix} \cdot \begin{bmatrix} -B_{kn} \\ G_{kn} \end{bmatrix} \tag{17}$$

In view of (9), they can be calculated as:

$$\frac{\partial \bar{f}_k}{\partial V_k^R} = \mathrm{vect}\left(\mathbf{I}_k + \mathbf{V}_k^* \, \mathbf{Y}_{kk}\right) \tag{18}$$

and

$$\frac{\partial \bar{f}_k}{\partial V_k^I} = \mathrm{vect}(-j\mathbf{I}_k + \mathbf{V}_k^* \, j\mathbf{Y}_{kk}). \tag{19}$$

As a conclusion, the calculation of the diagonal elements of Jacobian relies on the same voltage phasors-by-admittances multiplications as for the off-diagonal elements followed by the addition of node currents and further vector elements reordering.

Due to the relatively simple form of expressions (14) (15) (18) (19), their calculations can be easily implemented as array operations extended to all of the PQ type nodes. Consider, for instance, the task of calculating the complex multiplications $\mathbf{V}_k^* \, \mathbf{Y}_{kn}$ in (14) for all of the $N_{PQ} \leq N$ nodes of type PQ. To this aim, the indices of PQ type nodes can be collected into the vector index

$$\vec{k} = [i_1, i_2, \ldots, i_{N_{PQ}}]. \tag{20}$$

Hence, if $\vec{\mathbf{V}}$ is the column vector of the N complex voltage phasors, using the standard Matlab notation, we have that $\vec{\mathbf{V}}(\vec{k})$ denotes the vector collecting the subset of voltage phasors at the PQ nodes. As a preliminary elaboration, such voltages are complex conjugated and reordered as the diagonal elements of the following diagonal matrix

$$\mathbf{D}_{\vec{k}} = \mathrm{diag}(\mathrm{conj}(\vec{\mathbf{V}}(\vec{k}))) \tag{21}$$

In Matlab notation, we have that $\mathbf{Y}(\vec{k}, :)$ denotes the $N_{PQ} \times N$ complex submatrix formed by the N_{PQ} rows of node admittance matrix $\mathbf{Y}$ selected by vector index $\vec{k}$. In view of that, we conclude that the following *single array-based operation*

$$\mathbf{D}_{\vec{k}} \cdot \mathbf{Y}(\vec{k}, :) \tag{22}$$

provides the $N_{PQ} \times N$ partial derivatives $\mathbf{V}_k^* \, \mathbf{Y}_{kn}$ appearing in (14) for all of the PQ nodes versus all of the nodes of the grid. In a similar way,

$$j\mathbf{D}_{\vec{k}} \cdot \mathbf{Y}(\vec{k}, :) \tag{23}$$

supplies the $N_{PQ} \times N$ partial derivatives $\mathbf{V}_k^* j\mathbf{Y}_{kn}$ appearing in (15).

B. PV NODES

For the kth node of PV type, the power flow equations are modified as follows:

$$\begin{aligned} \mathrm{Re}(\mathbf{V}_k^* \, \mathbf{I}_k) - P_k &= 0 \\ |\mathbf{V}_k|^2 - V_m^2 &= 0 \end{aligned} \tag{24}$$

where V_m is the constrained voltage module. In vector form, equations (24) are rewritten as:

$$\begin{aligned} \bar{f}_k = &\begin{bmatrix} V_k^R & V_k^I \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} I_k^R \\ I_K^I \end{bmatrix} + \begin{bmatrix} -P_k \\ -V_m^2 \end{bmatrix} \\ &+ \begin{bmatrix} 0 & 0 \\ V_k^R & V_k^I \end{bmatrix} \cdot \begin{bmatrix} V_k^R \\ V_K^I \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{aligned} \tag{25}$$

along with (11) for currents. The off-diagonal elements of the Jacobian are

$$\frac{\partial \bar{f}_k}{\partial V_n^R} = \begin{bmatrix} V_k^R & V_k^I \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} G_{kn} \\ B_{kn} \end{bmatrix} \tag{26}$$

and

$$\frac{\partial \bar{f}_k}{\partial V_n^I} = \begin{bmatrix} V_k^R & V_k^I \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} -B_{kn} \\ G_{kn} \end{bmatrix}. \tag{27}$$

Using (9), we find:

$$\frac{\partial \bar{f}_k}{\partial V_n^R} = \begin{bmatrix} \mathrm{Re}(\mathbf{V}_k^* \, \mathbf{Y}_{kn}) \\ 0 \end{bmatrix} \tag{28}$$

and

$$\frac{\partial \bar{f}_k}{\partial V_n^I} = \begin{bmatrix} \mathrm{Re}(\mathbf{V}_k^* j \, \mathbf{Y}_{kn}) \\ 0 \end{bmatrix} \tag{29}$$

With a similar reasoning, the diagonal terms of the Jacobian matrix result:

$$\frac{\partial \bar{f}_k}{\partial V_k^R} = \begin{bmatrix} \mathrm{Re}(\mathbf{I}_k + \mathbf{V}_k^* \, \mathbf{Y}_{kk}) \\ 2V_k^R \end{bmatrix} \tag{30}$$

and

$$\frac{\partial \bar{f}_k}{\partial V_k^I} = \begin{bmatrix} \mathrm{Re}(-j\mathbf{I}_k + \mathbf{V}_k^* j \, \mathbf{Y}_{kk}) \\ 2V_k^I \end{bmatrix} \tag{31}$$

We conclude that also for PV nodes, the Jacobian matrix formation is dominated by complex multiplications $\mathbf{V}_k^* \mathbf{Y}_{kn}$ followed by numerically inexpensive vector reordering. Such multiplications can be implemented as a single array operation extended to all of the PV type nodes.

V. APPLICATIONS OF CAN METHOD

The robustness of NR solver (when applied to cartesian coordinates formulation) joined with the numerical efficiency of complex-array operations make the proposed CAN method suitable for those advanced simulations requiring a huge number of repeated power flow analyses. In what follows, we illustrate the application of CAN method to Probabilistic Power Flow (PPF) simulation. PPF simulations in modern power grids are often required since some parameters in the grids are affected by uncertainty [15], [16]. Such uncertainty is accounted for by a set of l random variables ξ_r that for notational compactness can be collected into a vector $\vec{\xi} = [\xi_1, \xi_2, \ldots, \xi_l]$. Mathematically, each ξ_r is a random variable described by the joint Probability Density Function (PDF) $\rho_r(\xi_1, \ldots, \xi_l)$ [17]. In practice, uncertainty parameters in a smart grid may be the values assumed at a given time or time window of some generated or absorbed powers at the grid terminations. For instance, the active (or reactive) power generated (or absorbed) at node kth in the grid can take the form

$$P_k = P_k^0 + \xi_k\, S_k P_k^0, \tag{32}$$

where P_k^0 is the nominally expected power while $\xi_k S_k\, P_k^0$ is the random fluctuation determined by a normalized random variable ξ_k scaling power fraction $S_k\, P_k^0$. In this case, due to the uncertainty in power generation, each observable variable describing the electrical state of the network, e.g. the magnitude of a node voltage $V = |\mathbf{V}_n|$, can be thought of as a random variable that depends on the uncertainties vector, i.e. $V(\vec{\xi}) = |\mathbf{V}_n(\vec{\xi})|$. The target of PPF is the determination of the PDF of the observable variable V and/or the evaluation of its mean value and interval of variability.

The most general and robust approach to PPF problem is that based on Monte Carlo (MC) method. In fact, even if approximate numerically more efficient methods exist to speed up the PPF analysis, such techniques always require validation via comparisons to MC.

Using MC method, the statistical description of the observable variable $V(\vec{\xi})$ is achieved by generating a very large number N_{mc} of uncertainty vectors $\vec{\xi}^1, \vec{\xi}^2, \ldots, \vec{\xi}^{N_{mc}}$ according to the joint probability distribution of variables in $\vec{\xi}$. For each realization $\vec{\xi}^n$ of the statistical variables, the realization of the iobservable quantity $V(\vec{\xi}^n)$, is determined by running one deterministic power flow simulation. As the number N_{mc} of evaluations grows, at limit tending to infinity, the distribution of the values $V(\vec{\xi}^n)$ tends to the PDF of V. However, due to the slow $1/\sqrt{N_{mc}}$ convergence rate of MC method, the number of repeated simulations actually needed to obtain an accurate approximation of the PDF can be quite large making probabilistic analysis very time consuming. To address such an issue, we explore the implementation of MC method in connection with the proposed numerically efficient CAN solver implemented in the array-operation supporting language Matlab. The main steps of MC implementation are highlighted in Algorithm1. One salient feature of the implementation is that the generation of random samples and power flow simulations are developed in the same mathematical framework. In this way, time expensive calls to general purpose simulation codes are avoided. Furthermore, the formation of the grid primitives (e.g. the node admittance matrix) is done only once thus achieving great numerical efficiency.

Algorithm 1 Steps of Monte Carlo Simulation

```
Array-operation framework:
Form power flow primitives (e.g. node admittance matrix)
Generate $N_{mc}$ samples $\vec{\xi}^n$ according to Joint PDF
For each $\vec{\xi}^n$ do
 Calculate termination powers $P_k, Q_k$ and load into power flow equations
            $\Downarrow$
 Power Flow Simulation:
 Run CAN solver
 Produce Output Results
            $\Downarrow$
 Determine $V(\vec{\xi}^n)$
EndFor
Using $N_{mc}$ simulated $V(\vec{\xi}^n)$, estimate PDF, mean value, variance
```

In order to further speed up the PPF analysis, we also consider combining the CAN solver with a MC acceleration technique based on Polynomial Chaos (gPC) expansions. The idea behind this method is that of approximating the deterministic relationship $V(\vec{\xi}) = |\mathbf{V}_n(\vec{\xi})|$ with an order-β truncated series of the type [18]

$$V(\vec{\xi}) \approx \sum_{i=1}^{N_b} c_i\, H_i(\vec{\xi}) \tag{33}$$

and use a limited set of simulations to estimate coefficients c_i. In (33), in fact $H_i(\vec{\xi})$ are N_b known multi-variate polynomial functions, whose expressions depend on the form of $\vec{\xi}$ joint PDF, while c_i are the unknown coefficients to be determined. An efficient way to calculate c_i is through a collocation-based method [19]. In its basic implementation, the method implies selecting $N_s = N_b$ testing points $\vec{\xi}^n$, for $n = 1, \ldots, N_s$ where the values of the desired observable variable $V^n = V(\vec{\xi}^n)$ are calculated running N_s deterministic Power Flow simulations. The coefficients c_i are deduced by enforcing the series expansion (33) to fit *exactly* (i.e., the polynomials interpolate the samples) the values V^n at the N_s testing points.

Using collocation method, for a given truncation order β and number of parameters l, the number of required simulations is [19]

$$N_s = \frac{(\beta + l)!}{\beta! \, l!}. \tag{34}$$

Once the coefficients c_i have been computed, the gPC expansion (33) provides a surrogate model of the $V(\vec{\xi})$ multidimensional dependence. Such a model can be exploited within the MC framework (in place of running many Power Flow analyses) to deduce the detailed PDF of the observable quantity $V(\vec{\xi})$ its mean value and standard deviation. For medium-size problems (e.g. with a number of statistical parameters < 50) and low truncation order $\beta \leq 3$, the gPC-based method can introduce a significant speed up compared to standard MC method. In this paper, we implemented the gPC method and use the CAN solver for simulating the testing points.

VI. NUMERICAL EXAMPLES

A. SINGLE PHASE GRID

In the first example we compare the CAN solver with the reference Newton-Raphson (NR) method implemented in the MatPower tool. This latter adopts a power flow problem formulation where node voltages are represented in polar coordinates.

To this aim we run some of the single-phase power flow cases, i.e. case69, case118, case145. case300, and case1354pegase, provided as a benchmark within the MatPower software suite. In both MatPower NR and CAN solvers, the unknown voltages $V_k = |V_k| e^{j\delta_k}$ are initialized with module and phase values that are sufficiently close to those of the final solution. Such initial values, which are available in the benchmark files, are determined via preliminary DC analyses or through an iterative solver used as initializer. As a result, for all of the considered cases, NR and CAN solvers converge in a few iterations (e.g. of the order of four or five). Table 1 reports the simulation times (including loading the equations and solving them) taken by MatPower NR and CAN to perform a single power flow analysis for the considered single phase cases. It is seen how CAN solver generally outperforms MatPower NR due to the exploitation of the inherent array-operation capability of Matlab tool.

TABLE 1. Simulation times [s].

	Case69	case118	Case145	Case300	Case1354
MatP NR	0.05	0.07	0.10	0.31	0.75
CAN	0.03	0.04	0.06	0.15	0.31

Then, in order to verify the robustness of the two methods considered, we focus on the case145 (this case refers to a meshed network with many PV nodes), and repeat the power flow analysis starting from generic random initializations of the NR method. More specifically, from the close-to-solution module values $|V_k|$ provided along with the benchmark files, we generate different initial guess as follows

$$|V_k|_{init} = |V_k| \pm A\, u \tag{35}$$

where A is the percentage perturbation amplitude and u is a random variable uniformly distributed in the interval $[-1, 1]$. One hundred sets of initial values are generated in this way and for each set the power flow analysis is repeated. Table 2 reports the number of successful power flow analyses achieved with MatPower NR and CAN methods. We see how increasing the randomization degree A of the initial guess, the number of successful MatPower analyses, formulated in polar coordinates, reduces significantly. By contrast, CAN method formulated in cartesian coordinates is more robust, in fact it preserves a one-hundred percent successful rate. It is worth observing how unsuccessful analyses are those where the nonlinear solver either diverges or converges to non-physical ill-conditioned solutions that contain low voltages values. As an example, in Fig. 3, it is shown (thick blue line) the voltage module values that are the physically-correct solution for the case145 power flow. In the same figure, it is reported one randomly generated initialization (yellow square marker), corresponding to $A = 0.10$, along with the associated non-physical solution (red triangular marker) obtained by running MatPower NR solver with such an initialization.

TABLE 2. Number of successful simulations.

	$A = 8\%$	$A = 10\%$	$A = 15\%$
MatP NR	80	61	< 10
CAN	100	100	100

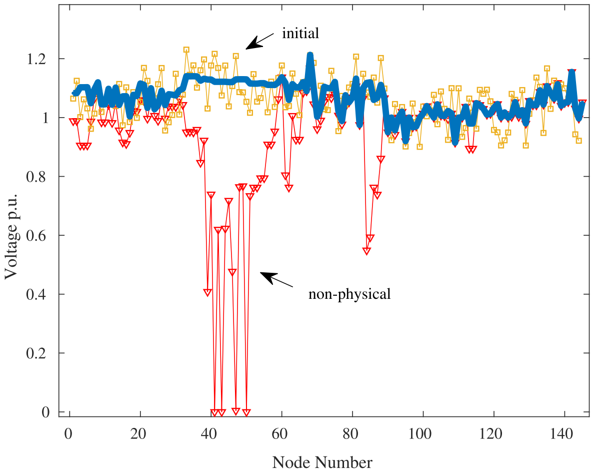

FIGURE 3. Node voltage modules for the case145 grid: (thick blue line) physically correct solution; (yellow square marker) initialization; (red triangular marker) non-physical solution provided by MatPower NR.

B. THREE-PHASE GRID

In this subsection, we focus on the IEEE 118 bus test case provided within the file case118.m of the MatPower suite. First, this single-phase test grid is extended to a balanced three-phase network, referred to as balanced 118bus grid, by replicating the single-phase loads and generators at the three nodes of each bus. Three-phase lines are modeled with the π circuit reported in Fig. 2 where the impedance matrix $\mathbf{Z}_{abc}$ in (7) is derived from the single phase impedance $\mathbf{Z}_s$. In fact, the diagonal terms are selected such that $\mathbf{Z}_a = \mathbf{Z}_b = \mathbf{Z}_c = \mathbf{Z}_s$ while coupling terms are $\mathbf{Z}_{ab} = \mathbf{Z}_{ac} = \mathbf{Z}_{bc} = \alpha\,\mathbf{Z}_s$ with $\alpha = 0.5$. Such a selection corresponds to a realistic line configuration with a significant coupling among them.

Second, from the balanced grid, an unbalanced three-phase case (referred to as unbalanced 118bus grid) is derived by perturbing a subset of eleven power generators. Table 3 lists such generators showing the bus number at which they are connected, the nominal active power they generate and the module of the node voltage they impose (PV nodes). These nominal active power values P_k are kept fixed at the nodes of Phase-B and Phase-C lines while they are perturbed at the nodes of Phase-A in order to create unbalance among the lines. The aim is that of exploring the effects that fluctuations in power generation P_k^A at Phase-A line can have on the voltage nodes of the three phase lines. As a preliminary result, we first perform a deterministic simulation where perturbed powers at Phase-A nodes are increased by a 20% factor as follows

$$P_k^A = P_k\,(1 + S) \tag{36}$$

where P_k are reported in Table 3 and $S = 0.2$. With these parameters, we simulate the three-phase balanced 118bus grid and the unbalanced one and compare results.

TABLE 3. Unbalanced generators.

Bus N.	P_k [p.u.]	$\lvert V_k\rvert$ [p.u.]
10	4.50	1.05
12	0.85	0.99
25	2.20	1.05
26	3.14	1.015
49	2.04	0.97
59	1.55	0.985
61	1.60	0.995
65	3.91	1.005
66	3.92	1.05
80	4.77	1.04
89	6.07	1.005

Fig. 4 shows the computed voltage magnitudes at the $118 \times 3 = 354$ nodes in the balanced and unbalanced grids. Fig. 5 reports a detail of such voltage magnitudes for a subset of nodes and, in particular, for the three nodes (ordered in Phase A, B and C) at bus 42 and 44. Nodes at Bus 42 are of PV type and thus their voltage magnitudes remain fixed at the regulated values ($|V_k| = 0.985$ V) in both balanced and unbalanced grids. By contrast, nodes at Bus 44 are of PQ type and thus their voltages are free to vary. In particular, for the assumed increase of generated powers, the Bus 44 Phase-A node voltage grows while Phase B and C voltages reduce. We checked the correctness of these results via comparisons with MatPower simulations (for the balanced grid) and with OpenDSS simulator (for the unbalanced grid). The differences among the (per unit) voltage values provided by the CAN solver compared to those obtained with MatPower and/or OpenDSSare are always $< 10^{-3}$.

Second, we perform a PPF analysis where the generated active power at Phase-A nodes of the eleven buses listed in Table 3 are perturbed as follows:

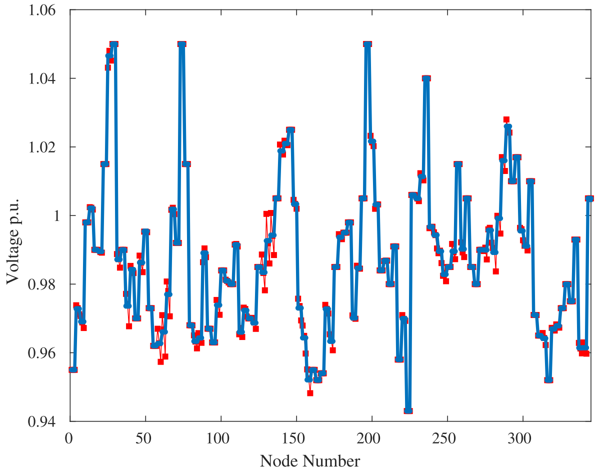

FIGURE 4. Voltage magnitudes: (blue thick line) in the balanced three-phase 118bus grid; (red square marker) in the unbalanced 118bus grid.

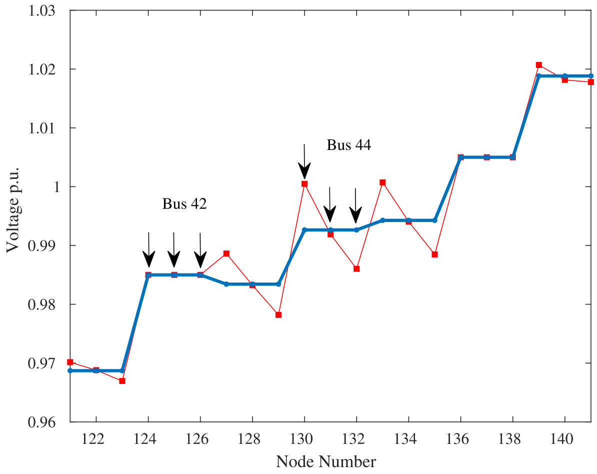

FIGURE 5. Detail of the node voltage magnitudes at Bus 42 and 44 for: (blue thick line) the balanced and (red square marker) unbalance 118bus grid.

$$P_k^A = P_k\,(1 + S\,\xi_k) \tag{37}$$

where ξ_k are independent and Normally distributed random variables (with zero mean and unitary variance) modeling uncertainty in power generation.

Probabilistic analysis is first performed in accordance to the steps described in Algorithm1 by running 10, 000 MC simulations with the proposed CAN solver within the Matlab framework. The MC+CAN overall simulation time is about 290 s on a i7 Quadcore computer. The same MC simulation performed calling from the Matlab workspace the general purpose OpenDSS simulator takes about 840 s thus resulting about three times slower.

Finally, as a further check, we repeat the PPF analysis by exploiting the acceleration technique based on generalized polynomial chaos (gPC) method and CAN solver. In our example, we have $l = 11$ random variables and we adopt expansion order $\beta = 3$. For this example, the number of testing points, i.e. the number of required power flow simulations, of the gPC method is 364. The speed up factor achieved by gPC method over MC method is about 27×.

Fig. 6 shows the PDFs of the voltage at Bus 44, Phase-A (i.e. node 130) calculated with the MC method and with the accelerated gPC method are reported in Fig. 6: it is seen how they match with great accuracy thus confirming the reliability of the gPC method. After its validation, the gPC method combined with CAN solver is further exploited for numerically efficient explorations of other relevant quantities in the unbalanced grid. Fig. 7, for instance, shows and compare the calculated PDFs for voltages of Phase A B and C at Bus 44. We can observe how Phase-A node voltage (i.e. the same phase line with generation power random fluctuations) is the one with the greatest variability, while Phase-B node voltage exhibits a much narrower variability interval. We also observe how Phase-B voltage is not Gaussian distributed due to the nonlinearity of power flow problem. As a final example, in Fig. 8 we show and compare the PDFs of the Phase A, B and C voltages in another PQ-type Bus, i.e. Bus 21 in the grid. Interestingly at Bus 21, Phase-C node voltage exhibits a variability interval that is greater than that of Phase-A voltage. This result, that we further checked with reference MC simulations, is due to the complex interaction between delivered and absorbed powers in the unbalanced 118bus grid and to phase lines coupling. Such an example highlights how electrical variables variability in an unbalanced grid can hardly be predicted *a priori* and how accurate probabilistic simulations are indispensable.

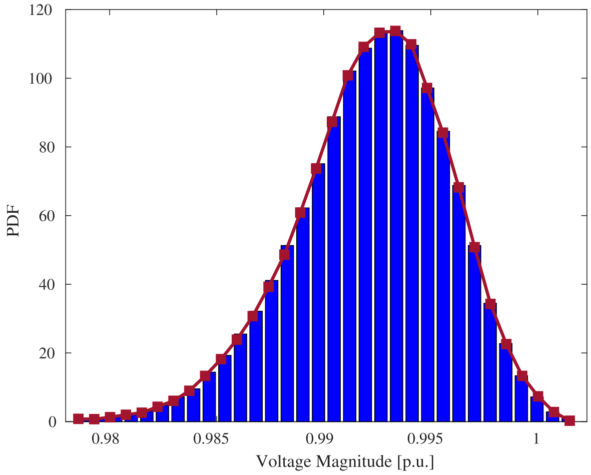

FIGURE 6. PDF of voltage at Bus 44 Phase A calculated with: (bar histogram) MC method; (red square marker) gPC method.

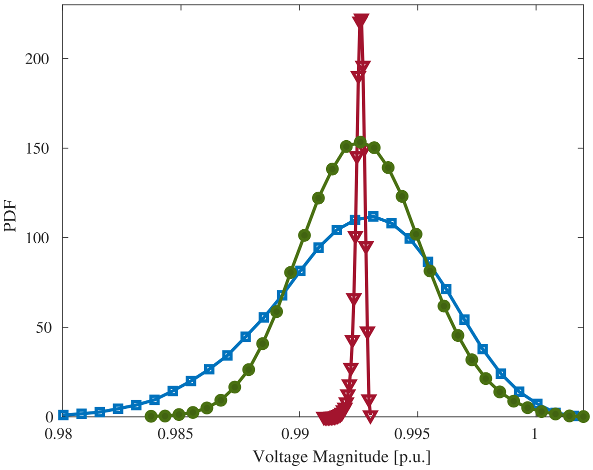

FIGURE 7. PDFs of voltages at Bus 44: (blue square marker) Phase A; (read triangle marker) Phase B (this PDF is divided by 10 for scale reasons); (green circle marker) Phase C.

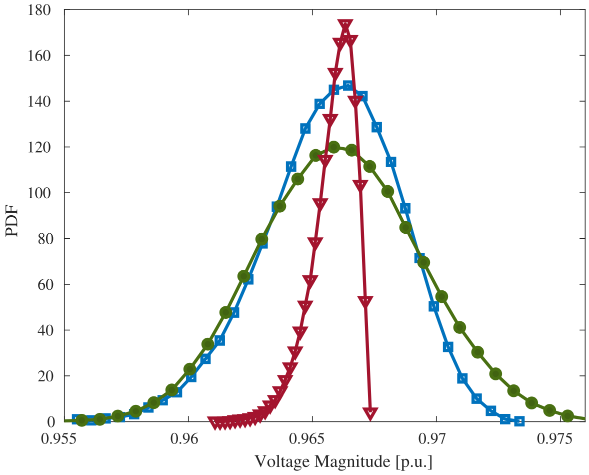

FIGURE 8. PDFs of voltages at Bus 21: (blue square marker) Phase A; (read triangle marker) Phase B (this PDF is divided by 3 for scale reasons); (green circle marker) Phase C.

VII. CONCLUSION

In this paper, we have described and original vector-based formulation of the power flow equations in a power grid. Such a formulation allows the implementation of the robust and general NR solver by means of elementary array operations developed in the complex numbers field. An efficient version of the method, referred to as CAN method, has been implemented in the Matlab language by exploiting its complex-array operation capability. We have described how the CAN method can be exploited in probabilistic simulations in connection to standard MC method as well as with acceleration techniques based on generalized polynomial chaos expansions. The efficiency and robustness of the novel simulation technique have been checked via comparisons with standard simulations tools, i.e. MatPower and OpenDSS, in the analysis of benchmark single-phase network as well as in unbalanced three-phase grids with uncertain power generators.

REFERENCES

[1] F. Vallée, V. Klonari, T. Lisiecki, O. Durieux, F. Moiny, and J. Lobry, “Development of a probabilistic tool using Monte Carlo simulation and smart meters measurements for the long term analysis of low voltage distribution grids with photovoltaic generation,” *Int. J. Electr. Power Energy Syst.*, vol. 53, pp. 468–477, Dec. 2013.

[2] M. Neaimeh, R. Wardle, A. M. Jenkins, J. Yi, G. Hill, P. F. Lyons, Y. Hübner, P. T. Blythe, and P. C. Taylor, "A probabilistic approach to combining smart meter and electric vehicle charging data to investigate distribution network impacts," *Appl. Energy*, vol. 157, pp. 688–698, Nov. 2015.

[3] H. Mortazavi, H. Mehrjerdi, M. Saad, S. Lefebvre, D. Asber, and L. Lenoir, "A monitoring technique for reversed power flow detection with high PV penetration level," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2221–2232, Sep. 2015.

[4] K. Jhala, B. Natarajan, and A. Pahwa, "Probabilistic voltage sensitivity analysis (PVSA)—A novel approach to quantify impact of active consumers," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2518–2527, May 2018.

[5] J. Tang, F. Ni, F. Ponci, and A. Monti, "Dimension-adaptive sparse grid interpolation for uncertainty quantification in modern power systems: Probabilistic power flow," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 907–919, Mar. 2016.

[6] G. Gruosso, G. S. Gajani, Z. Zhang, L. Daniel, and P. Maffezzoni, "Uncertainty-aware computational tools for power distribution networks including electrical vehicle charging and load profiles," *IEEE Access*, vol. 7, pp. 9357–9367, 2019.

[7] T. Shu, X. Lin, S. Peng, X. Du, H. Chen, F. Li, J. Tang, and W. Li, "Probabilistic power flow analysis for hybrid HVAC and LCC-VSC HVDC system," *IEEE Access*, vol. 7, pp. 142038–142052, 2019.

[8] D. Shirmohammadi, H. W. Hong, A. Semlyen, and G. X. Luo, "A compensation-based power flow method for weakly meshed distribution and transmission networks," *IEEE Trans. Power Syst.*, vol. PWRS-3, no. 2, pp. 753–762, May 1988.

[9] J. Liu, M. M. A. Salama, and R. R. Mansour, "An efficient power flow algorithm for distribution systems with polynomial load," *Int. J. Electr. Eng. Edu.*, vol. 39, no. 4, pp. 371–386, Oct. 2002.

[10] W.-M. Lin, T.-S. Zhan, and M.-T. Tsay, "Multiple-frequency three-phase load flow for harmonic analysis," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 897–904, May 2004.

[11] H. Li, H. Zhou, T. Liu, and Q. Chen, "A loop-analysis theory based linear power flow method for three-phase distribution power system," *IEEE Access*, vol. 7, pp. 157389–157400, 2019.

[12] A. Pandey, M. Jereminov, M. R. Wagner, D. M. Bromberg, G. Hug, and L. Pileggi, "Robust power flow and three-phase power flow analyses," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 616–626, Jan. 2019.

[13] Z. A. Memon, R. Trinchero, Y. Xie, F. G. Canavero, and I. S. Stievano, "An iterative scheme for the power-flow analysis of distribution networks based on decoupled circuit equivalents in the phasor domain," *Energies*, vol. 13, no. 2, p. 386, 2020.

[14] L. Powell, *Power System Load Flow Analysis* (Electrical Engineering). New York, NY, USA: McGraw-Hill, 2004. [Online]. Available: https://books.google.it/books?id=2B2RZZcLMwwC

[15] M. Hajian, W. D. Rosehart, and H. Zareipour, "Probabilistic power flow by Monte Carlo simulation with Latin supercube sampling," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1550–1559, May 2013.

[16] Z. Ren, W. Li, R. Billinton, and W. Yan, "Probabilistic power flow analysis based on the stochastic response surface method," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2307–2315, May 2016.

[17] A. Papoulis and U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. New York, NY, USA: McGraw-Hill, Nov. 2001.

[18] D. Xiu and G. E. Karniadakis, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM J. Sci. Comput.*, vol. 24, no. 2, pp. 619–644, Jan. 2002.

[19] Z. Zhang, T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "Stochastic testing method for transistor-level uncertainty quantification based on generalized polynomial chaos," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 32, no. 10, pp. 1533–1545, Oct. 2013.

PAOLO MAFFEZZONI (Senior Member, IEEE) received the Laurea degree *(summa cum laude)* in electrical engineering from the Politecnico di Milano, Italy, in 1991, and the Ph.D. degree in electronic instrumentation from the Università di Brescia, Italy, in 1996. He is currently a Full Professor of electrical engineering with the Politecnico di Milano. His current research interests include modeling and simulation of nonlinear circuits and systems, oscillatory devices, stochastic simulation, and unconventional computing with applications to analog and mixed-signal electronics and power systems. He has published about 140 research articles in international journals and conferences. He has served as an Associate Editor for the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems and the IEEE Transactions on Circuits and Systems I: Regular Papers.

GIAMBATTISTA GRUOSSO (Senior Member, IEEE) was born in 1973. He received the M.S. and Ph.D. degrees in electrical engineering from the Politecnico di Torino, Italy, in 1999 and 2003, respectively. From 2002 to 2011, he was an Assistant Professor with the Department of Electronics and Informatics, Politecnico di Milano. Since 2011, he has been an Associate Professor with the Politecnico di Milano. His main research topics are electrical vehicles transportation electrification electrical power systems optimization and simulation of electrical systems. He is the author of more than 80 articles in journals and conferences on these topics.