Power Flow Analysis under Uncertainty using Symmetric Fuzzy Arithmetic

Manuel Marin, David Defour, Federico Milano

Power Flow Analysis under Uncertainty using Symmetric Fuzzy Arithmetic

Manuel Marin, David Defour, and Federico Milano, *Senior Member, IEEE*

***Abstract*—This paper present a novel approach using symmetric fuzzy numbers to deal with uncertainty in power system analysis. Special attention is put on reducing overestimation due to conservatism inherent in fuzzy arithmetic. The midpoint-radius interval representation and a linear programming optimization problem are used in order to enhance efficiency and accuracy, respectively. The proposed method is tested over the IEEE 14-bus test system and compared with other techniques available in the literature. The case study that the proposed method yields accurate results at a reduced computational cost.**

I. Introduction

UNCERTAINTY is inherent to any physical systems. This is particularly true for power systems, for which uncertainty can have several causes, e.g., imprecise demand forecast, price variability, renewable energy generation, economic growth, industry placement, and line ageing [1], [2]. Failing to properly account for uncertainties can, in some cases, lead to erroneous estimations or unsecure operating conditions. Therefore, a reliable tool to handle several possible scenarios and combinations of scenarios is crucial to provide a clear understanding of the expected behavior of the system. This paper focuses on how to properly account for uncertainties in the power flow problem, which is the most important tool for the steady-state analysis of power systems.

Conventional power flow analysis provides accurate results based on a single (deterministic) operating point [3]. Clearly, it is possible to solve multiple instances of the power flow problem for several (typically randomly generated) possible scenarios. This yields to the well-known Monte Carlo method. However, the computational burden of the Monte Carlo approach can be unsuitable for practical purposes, real-time analysis and preventive and/or corrective control actions [4].

In the past, two main solutions have been used to address the inclusion of uncertainty in power flow data. On one hand, we find methods based on probabilistic theory [5]–[8]. On the other hand, methods based on alternative arithmetic (e.g., interval, fuzzy, affine arithmetic) have been proposed [1], [9]–[11]. In both cases, Monte Carlo's simulations have been used as a validation tool.

The probabilistic power flow approach consists of using random variables to model the system; then, by applying specific results from theory of probability, a relation is established between the distribution of input parameters and output variables. The probability distribution of the inputs has to be determined first, upon statistical data. In [5], a linear dependence between line flows and nodal injections is assumed to determine relatively simple expressions for the former. This result is extended to include correlation between nodal injected powers in [6]. In [7], the authors present a novel approach that considers multiple linearization points, thus improving the accuracy of the solution, while incrementing the complexity of the algorithm. Detractors of the probabilistic power flow approach criticize the difficulty of determining accurate probability distributions of the inputs, especially if the data are scarce or do not follow any particular distribution [10].

Interval and fuzzy power flow methods attempt to overcome the latter issue through replacing the probability distribution by a *possibility distribution*, which can be defined based on experience and historical data. Sentences such as "load between 0.5 and 1 pu" and "generation around 0.9 pu" can be easily translated into intervals. Then, these intervals are manipulated according to the rules of interval or fuzzy arithmetic to compute interval results. In [9], the interval Newton's method is directly applied to solve a case of power flow analysis for a network of 5 buses, assuming small uncertainty in the nodal injected powers. In [10], a linearized approach is applied to compute fuzzy extensions of a *crisp* power flow solution, for several levels of uncertainty. In [11], an approach using linear programming, where the restrictions are obtained from fuzzy variables, is presented. While interval and fuzzy arithmetic have a low computational burden with respect to probabilistic methods, a major drawback is the risk to overestimate the intervals of the solution, especially in case the input parameters are characterized by wide interval. Clearly, large intervals can make the solution either of little practical interest or useless.

The objective of the present work is to develop a tool for power flow analysis under uncertainty, keeping a balance between low computational burden and accuracy of the results. With this aim, the paper presents a power flow analysis based on fuzzy arithmetic and proposes a linear programming problem that is able to overcome the main drawback of fuzzy arithmetic, i.e., overestimation of the solution intervals. Specific contributions of this paper are the following.

1) *Power flow solution under uncertainty.* We present a novel technique for power flow analysis under uncertainty, based on the direct application of the rules of fuzzy algebra to the standard power flow problem. The variables increment is carefully calculated at each step of

This work was conducted in the Electricity Research Centre, University College Dublin, Ireland, which is partly supported by the Commission for Energy Regulation, Bord Gáis Energy, Bord na Móna Energy, Cylon Controls, EirGrid, Electric Ireland, Energia, EPRI, ESB International, ESB Networks, Gaelectric, Intel, SSE Renewables, and UTRC. This publication has emanated from research conducted with the financial support of Actility.

M. Marin and D. Defour are with Université de Perpignan Via Domitia, DALI and Université Montpellier 2, LIRMM, France (e-mails: manuel.marin@univ-perp.fr and david.defour@univ-perp.fr).

F. Milano is with the School of Electrical, Electronic and Communications Engineering of the University College Dublin, Dublin, Ireland (e-mail: federico.milano@ucd.ie).

an iterative procedure, in order to minimize overestimation of the solution, a known issue in interval analysis. We compare our technique in terms of accuracy and complexity with two other fuzzy power flow methods, from the literature [9], [10].

2) *Optimized interval representation.* By making an assumption over the shape of the possibility distribution of fuzzy variables, we are able to effectively combine the midpoint-radius representation with the α-cut method, in order to simplify the basic operations of fuzzy linear algebra. We show that power flow analysis under uncertainty belongs to the class of application that can benefit of the computational advantages of the novel representation described in [12].

The remainder of the article is organized as follows. In Section II we introduce definitions and properties related to interval arithmetic used by the proposed methods which we detail in Section III. Results of simulation and comparisons with two others methods are presented in Section IV. Conclusions are drawn in Section V.

II. Interval and fuzzy linear algebra

In this section we present the notation used through the paper and most relevant rules and properties related to interval and fuzzy arithmetic.

Notations: In the remainder of this article, normal fonts (A) will denote crisp variables, bold fonts ($\boldsymbol{A}$) interval variables and tilde ($\tilde{\boldsymbol{A}}$) fuzzy variables. All operations and relations expressed over crisp matrices (e.g. '$\leq$', '$||$') are applied component-wise.

A. Interval analysis

Interval analysis is used to manipulate uncertainty in numerical methods. Variables are expressed as ranges of possibilities and operate under specific rules, ensuring that all the uncertainty in the operands is in the result.

Definition 1 (Midpoint-radius interval matrix). *Let $A_m, A_r \in \mathbb{R}^{n \times n}$, with $A_r \geq 0$. The interval matrix $\boldsymbol{A} = \langle A_m, A_r \rangle$ is the set given by*

$$\boldsymbol{A} = \{A \in \mathbb{R}^{n \times n} : |A_m - A| \leq A_r\}.$$

Definition 2 (Interval magnitude). *Let $\boldsymbol{A}$ be an interval matrix. The magnitude $|\boldsymbol{A}|$ is the point matrix given by*

$$|\boldsymbol{A}| = max\{|A|, A \in \boldsymbol{A}\}.$$

Or, equivalently,

$$|\boldsymbol{A}| = |A_m| + A_r.$$

Definition 3 (Interval algebraic operations). *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be interval matrices. The interval matrix $\boldsymbol{C} = \boldsymbol{A} \oplus \boldsymbol{B}$ is the set given by*

$$\boldsymbol{C} = \{A \oplus B : A \in \boldsymbol{A}, B \in \boldsymbol{B}\}.$$

In particular, for $\boldsymbol{A} = \langle A_m, A_r \rangle$, $\boldsymbol{B} = \langle B_m, B_r \rangle$ and $\oplus \in \{+, -, \cdot\}$, we have

$$\begin{aligned} \boldsymbol{A} + \boldsymbol{B} &= \langle A_m + B_m, A_r + B_r \rangle, \\ \boldsymbol{A} - \boldsymbol{B} &= \langle A_m - B_m, A_r + B_r \rangle, \\ \boldsymbol{A} \cdot \boldsymbol{B} &= \langle A_m B_m, (|A_m| + A_r)B_r + A_r|B_m| \rangle. \end{aligned}$$

1) System of interval linear equations: In classical interval analysis, interval bounds for the zeros of non-linear functions are computed using interval Newton methods [13]. These methods require to compute an interval enclosure of the solution of $\boldsymbol{A}x = \boldsymbol{b}$, where $\boldsymbol{A}$ is an interval matrix and $\boldsymbol{b}$ is an interval vector. The solution is given by

$$\Sigma(\boldsymbol{A}, \boldsymbol{b}) = \{x : \forall b \in \boldsymbol{b}, \forall A \in \boldsymbol{A}, Ax = b\}.$$

In other words, for each $b \in \boldsymbol{b}$ and $A \in \boldsymbol{A}$, there is $x \in \Sigma(\boldsymbol{A}, \boldsymbol{b})$ such that $Ax = b$. The interval *hull* of $\Sigma(\boldsymbol{A}, \boldsymbol{b})$ is denoted $\Box\Sigma(\boldsymbol{A}, \boldsymbol{b})$ and corresponds to the smallest interval containing $\Sigma(\boldsymbol{A}, \boldsymbol{b})$.

For the proposed method, we do not rely on Newton method. We evaluate a specific subset $\Delta(\boldsymbol{A}, \boldsymbol{b})$ of $\Box\Sigma(\boldsymbol{A}, \boldsymbol{b})$, defined as follows.

Definition 4 ($\Delta(\boldsymbol{A}, \boldsymbol{b})$). *Let $\boldsymbol{A}$ be an interval matrix, $\boldsymbol{b}$ and $\boldsymbol{x}$ interval vectors. Then, $\Delta(\boldsymbol{A}, \boldsymbol{b})$ is the interval vector given by*

$$\Delta(\boldsymbol{A}, \boldsymbol{b}) = min\{\boldsymbol{x} : \boldsymbol{b} \subseteq \boldsymbol{A}\boldsymbol{x}\},$$

with the function min operating on magnitude (Def. 2).

In this case, for each $b \in \boldsymbol{b}$, there is $A \in \boldsymbol{A}$ and $x \in \Delta(\boldsymbol{A}, \boldsymbol{b})$ such that $Ax = b$. All the variability in $\boldsymbol{b}$ is included in this set, even if it is only a subset of $\Box\Sigma(\boldsymbol{A}, \boldsymbol{b})$. One can notice that $\Delta(\boldsymbol{A}, \boldsymbol{b})$ has the minimal magnitude.

In order to compute $\Delta(\boldsymbol{A}, \boldsymbol{b})$, we use the rules of interval algebra to rewrite the condition $\boldsymbol{b} \subseteq \boldsymbol{A}\boldsymbol{x}$ as follow:

$$A_m x_m = b_m, \tag{1a}$$

$$(|A_m| + A_r)x_r + A_r|x_m| \geq b_r, \tag{1b}$$

$$x_r \geq 0. \tag{1c}$$

Equation (1a) is a crisp linear system. It can be solved according to x_m and used in equation (1b). Then, it remains to find a minimal x_r that satisfies (1b) and (1c). Solving the following linear program can do this:

$$\begin{aligned} &\text{Minimize} && \|x_r\|_1 , \\ &\text{subject to} && (|A_m| + A_r)x_r \geq b_r - A_r|x_m|, \\ & && x_r \geq 0, \end{aligned} \tag{2}$$

with $\|x_r\|_1$ the sum of all the components of x_r. The program (2) is always feasible and its solution can be computed with any linear solver.

B. Fuzzy analysis

Whereas interval analysis allows accounting for one single level of uncertainty, fuzzy analysis accounts for several levels of uncertainty simultaneously.

Definition 5 (Fuzzy matrix). *Let $\mu_{\tilde{\boldsymbol{A}}} : \mathbb{R}^{n \times n} \to [0, 1]$. The fuzzy matrix $\tilde{\boldsymbol{A}}$ is the set of pairs given by*

$$\tilde{\boldsymbol{A}} = \{(A, \mu_{\tilde{\boldsymbol{A}}}(A)) : A \in \mathbb{R}^{n \times n}\}.$$

In the above definition, $\mu_{\tilde{\boldsymbol{A}}}$ is called the membership function and represents the grade of membership of each matrix $A \in \mathbb{R}^{n \times n}$ to $\tilde{\boldsymbol{A}}$.

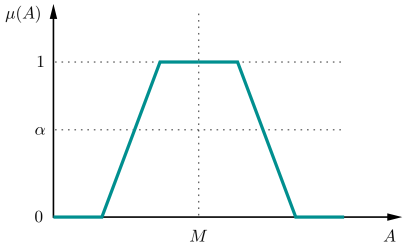

Fig. 1. Trapezoidal, symmetric membership function.

Figure 1 shows an example membership function for a 1×1 fuzzy matrix over real numbers.

Each $\alpha \in [0, 1]$ represents a level of uncertainty. For each level, we can obtain an α-cut of the fuzzy matrix which corresponds to the interval of variability in that level.

Definition 6 (α-cut)**.** *Let $\tilde{\boldsymbol{A}}$ be a fuzzy matrix with membership function $\mu_{\tilde{\boldsymbol{A}}}$. Let $\alpha \in [0, 1]$. The α-level cut (also, α-cut) of $\tilde{\boldsymbol{A}}$ is the set of real matrices given by*

$$\boldsymbol{A}_\alpha = \{A \in \mathbb{R}^{n\times n} : \mu_{\tilde{\boldsymbol{A}}}(A) \geq \alpha\}.$$

Note that this is an interval matrix. As an interval matrix, it can be manipulated using the rules of interval algebra. This allows us to introduce the α-cut method for fuzzy algebra.

Definition 7 (α-cut method)**.** *Let $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$ and $\tilde{\boldsymbol{C}}$ be fuzzy matrices. Let $\alpha \in [0, 1]$ and $\boldsymbol{A}_\alpha$, $\boldsymbol{B}_\alpha$ and $\boldsymbol{C}_\alpha$ be the corresponding α-cuts. Then*

$$\tilde{\boldsymbol{C}} = \tilde{\boldsymbol{A}} \oplus \tilde{\boldsymbol{B}} \iff \boldsymbol{C}_\alpha = \boldsymbol{A}_\alpha \oplus \boldsymbol{B}_\alpha.$$

In other words, operations involving fuzzy matrices are done by splitting fuzzy matrices according to their α-cuts. Next, interval operations are performed between α-cuts of the same level to obtain the corresponding α-cuts of the result.

It was shown that fuzzy arithmetic can benefit, in terms of memory footprint and number of operations, from the midpoint-radius representation when fuzzy numbers are symmetric [12]. This corresponds to the case where the midpoint of any α-cut is independent of α. The same can apply for symmetric fuzzy matrices defined as follows:

Definition 8 (Symmetric fuzzy matrix)**.** *Let $\tilde{\boldsymbol{A}}$ be a fuzzy matrix and $M \in \mathbb{R}^{n\times n}$. Let $\mu_{\tilde{\boldsymbol{A}}}$ be symmetric around M, i.e.*

$$\mu_{\tilde{\boldsymbol{A}}}(M - A) = \mu_{\tilde{\boldsymbol{A}}}(M + A), A \in \mathbb{R}^{n\times n}.$$

Then $\tilde{\boldsymbol{A}}$ is a symmetric fuzzy matrix.

Therefore, algebraic operations between symmetric fuzzy matrices represented with midpoint-radius correspond to, first evaluate the midpoint, which is common to all the α-cuts, and then compute the radius independently for each α-cut. This notably simplifies the definition and implementation of fuzzy operations.

III. Fuzzy power flow

In this section we describe a new method for power flow analysis under uncertainty. For the sake of completeness, the standard crisp power flow equations are recalled first.

A. Crisp power flow analysis

Crisp power flow analysis allows computing the balanced state of a system when uncertainty is not considered. The voltage magnitudes and real power injections at generator buses, as well as the real and reactive power injections at load buses, are given, crisp values. These data are used to compute the complex voltage at each bus, by solving the following power flow equations:

$$\begin{aligned} p_i &= \sum_{h\in\mathcal{B}} [e_i(g_{ih}e_h - b_{ih}f_h) + f_i(g_{ih}f_h + b_{ih}e_h)], i \in \mathcal{B} \\ q_i &= \sum_{h\in\mathcal{B}} [f_i(g_{ih}e_h - b_{ih}f_h) - e_i(g_{ih}f_h + b_{ih}e_h)], i \in \mathcal{B}_{\text{PQ}} \\ v_i^2 &= e_i^2 + f_i^2, i \in \mathcal{B}_{\text{PV}}. \end{aligned} \tag{3}$$

where p_i and q_i are the active and reactive power injections at bus i, respectively; $\bar{v}_i = e_i + jf_i$ is the bus voltage phasor at bus i; $\bar{y}_{ih} = g_{ih} + jb_{ih}$ is the (i, h) element of the network admittance matrix; $\mathcal{B} = 1, 2, \ldots, n$ is the set of network buses; and $\mathcal{B}_{\text{PQ}}$ and $\mathcal{B}_{\text{PV}}$ are the sets of PQ load and PV generator buses, respectively. Finally, the slack bus is defined as follows:

$$e_i = v_{\text{slack}}, \qquad f_i = 0 \tag{4}$$

Equations (3) and (4) can be conveniently rewritten using the following compact notation:

$$c = g(y) \tag{5}$$

where c is the vector of assigned input data, i.e., p_i and q_i at PQ buses and v_i^2 at PV buses; and y is the vector of unknowns, i.e., e_i and f_j at all buses but the slack one.

Note that we use rectangular coordinates to represent bus voltage phasors. This representation format avoid trigonometric function evaluations that are difficult to handle with symmetric fuzzy numbers. One can observe also that the evaluation of (5) solely implies additions, subtractions and multiplications, which are well defined and relatively inexpensive when using the midpoint-radius interval representation.

B. Fuzzy power flow analysis

Extending each constant and variable of (5) from a crisp representation to a fuzzy representation allows considering multiple levels of uncertainty. We group all the left hand sides into a constant fuzzy vector $\tilde{\boldsymbol{c}}$, and all the right hand sides into a variable fuzzy vector $\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}})$, where $\tilde{\boldsymbol{x}}$ is also a fuzzy vector. The corresponding system is given by:

$$\tilde{\boldsymbol{c}} = \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}). \tag{6}$$

In the case of fuzzy analysis, the equality of the equation (6) corresponds to a condition for the optimal solution; since any $\tilde{\boldsymbol{x}}$ satisfying

$$\tilde{\boldsymbol{c}} \subseteq \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}) \tag{7}$$

is also a solution of the system, that is, a set of states that accounts for all the variability expressed in the external inputs.

The algorithm we depict next takes $\tilde{\boldsymbol{c}}$, $\tilde{\boldsymbol{g}}$ and $\tilde{\boldsymbol{J}}$, the fuzzy Jacobian matrix of $\tilde{\boldsymbol{g}}$, as inputs, and produces $\tilde{\boldsymbol{x}}^*$, a solution of (7) 'close' to the solution of (6). We use the subscript α to designate α-cuts. Also, we assume that all the fuzzy elements are symmetric, so we can use the subscript m to designate the midpoints and r,α to designate the radius of each α-cut.

The algorithm proceeds as follows.

1) Solve the crisp power flow, defined by the system of non-linear equations $g_m(x) - c_m = 0$ by any crisp power flow analysis method, (i.e. Newton-Rahpson method). Let $x^{(0)}$ be the solution.

2) Compute $\tilde{\boldsymbol{x}}^{(0)}$, the initial guess for the fuzzy power flow, by
$$\boldsymbol{x}_\alpha^{(0)} = \langle x^{(0)}, 0 \rangle,$$
for all α. Set $k = 0$.

3) If $\tilde{\boldsymbol{c}} \subseteq \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}^{(k)})$, set $\tilde{\boldsymbol{x}}^* = \tilde{\boldsymbol{x}}^{(k)}$ and stop. We have found a solution. Otherwise, continue to step 4.

4) Compute $\tilde{\boldsymbol{f}}^{(k)} = \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}^{(k)}) - \tilde{\boldsymbol{c}}$, the fuzzy equation mismatches.

5) Compute $\tilde{\boldsymbol{J}}^{(k)} = \tilde{\boldsymbol{J}}(\tilde{\boldsymbol{x}}^{(k)})$, the fuzzy Jacobian matrix.

6) Compute $\boldsymbol{\Delta}\tilde{\boldsymbol{x}}$, the fuzzy variable increment, by
$$\boldsymbol{\Delta}\tilde{\boldsymbol{x}}_\alpha = \Delta(\tilde{\boldsymbol{J}}_\alpha^{(k)}, -\tilde{\boldsymbol{f}}_\alpha^{(k)}),$$
for all α.

7) Set $\tilde{\boldsymbol{x}}^{(k+1)} = \tilde{\boldsymbol{x}}^{(k)} + \boldsymbol{\Delta}\tilde{\boldsymbol{x}}$. Set $k = k + 1$ and go to step 3.

With respect to the issue of overestimation, the most critical step of the proposed algorithm is the calculation of the fuzzy variable increment $\boldsymbol{\Delta}\tilde{\boldsymbol{x}}$, in step 6. The usage of the Δ function as defined in Section II, ensures that all the variability expressed by the fuzzy mismatches is taken into account, with a minimal length optimization step. In other words, we progress towards the solution, by following optimization directions within the current solution interval. However, a minimal step is taken to avoid the classical drawback with interval arithmetic, which is the overestimation of the solution.

Figure 2 illustrates the convergence of the algorithm for the one-dimensional case. We use subscripts l and m to denote lower and upper bounds of the intervals. At the beginning we are at x^0, the crisp solution. From here we have only one direction to choose, which is the Jacobian on that point (blue tangent line). We find the interval $\boldsymbol{x^1} = [x_l^1, x_u^1]$. Now, we have a set of possible directions to choose from, corresponding to the Jacobian evaluated on all the points within the interval $\boldsymbol{x^1}$. We find that the Jacobian has the least magnitude when evaluated on the extreme points (orange tangent lines). We follow this direction and find the solution $\boldsymbol{x^*}$. Function g evaluated over the interval $\boldsymbol{x^*}$ contains all the points in the interval $\boldsymbol{c}$.

IV. Tests and results

In this section we present and discuss the results of our method on a test scenario. For the sake of comparison, we refer to two methods from literature: the standard fuzzy power flow method, described in [10], and the interval power flow method described in [9]. In the remainder of this section, we refer to these methods as method A and method B, respectively.

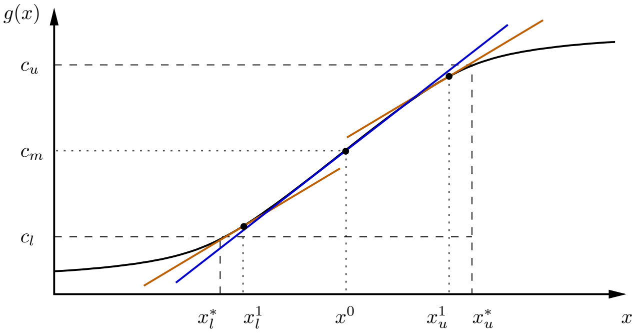

Fig. 2. Convergence of the algorithm in the one-dimensional case.

TABLE I
Input variables radius, in terms of percentage variation, for each α-level.

Variable name	$\alpha = 1$	$\alpha = 0$
Voltage magnitude	1	1.5
Real power injection	4	50
Reactive power injection	4	50

We build our test scenario by fuzzifying the well-known IEEE 14-bus system [14]. All variables are turned into fuzzy elements except for line impedances. We assume a trapezoidal, symmetric membership function for all the variables, which allows us to consider only two levels of uncertainty, $\alpha = 0$ (most uncertainty) and $\alpha = 1$ (least uncertainty).

Input variables, namely, voltage magnitudes and real power injections at generator buses and real and reactive power injections at load buses, are defined by setting the midpoint at the crisp value given by the IEEE network specification. The radius is defined by a percentage variation, depending on the type of variable, for each α-level. These percentages are summarized in Table I. They are chosen to mimic simulation results of method A and B.

The proposed method was implemented in C++ using the Coin-Clp library to solve the linear program [15]. The simulation of the described test scenario took 2.81 milliseconds on a Intel Core i5 CPU at 2.7 GHz, running Ubuntu Linux 12.10 64 bits.

A. Accuracy

Table II shows the voltage radius as a percentage of the midpoint value for each load bus, obtained by our method and by method A. The results of method A are taken from [16], where authors evaluate their method on the IEEE 14-bus system, considering a 50% variation over the crisp power injections. This corresponds to a scenario of uncertainty similar to α=0 (however, one can notice that method A does not allow to consider uncertainty in the generator voltage magnitudes). Out of this table, we can observe that the proposed method is in average 5 times more accurate. This can be the consequences of the following factors:

- Method A performs a single optimization step by moving from the crisp solution to a fuzzy solution. The length of this step is determined by the Jacobian of the last iteration of the Newton-Raphson method, used to solve the crisp power flow at a first stage of the process. In contrast, our

TABLE II
VOLTAGE MAGNITUDES RADIUS, IN TERMS OF PERCENTAGE VARIATION, IN OUR METHOD AND METHOD A.

Bus	**Our method**		**Method A**
	$\alpha = 1$	$\alpha = 0$	$\alpha = 0$
V4	1.385	1.910	4.39
V5	1.149	1.832	4.20
V7	0.963	1.422	8.53
V9	1.658	2.645	9.76
V10	1.310	1.828	10.35
V11	0.960	1.592	10.37
V12	1.065	1.744	10.96
V13	0.958	1.606	11.26
V14	1.529	1.706	11.92

method determines the smallest optimization step within an interval of possibilities, thus converging more slowly, but without overestimating the results as much as with method A.

- By forcing the generator voltage magnitudes to a crisp value, method A imposes much more variability on the rest of the voltages. The 1.5% variation assumed for the generator voltages when running the proposed method, allows accounting for some of the uncertainty introduced by the fuzzy power injections. Therefore, our method is more accurate than method A.

As for method B, we have no information on the results over the IEEE 14-bus system. In [9], the authors only study a 5-bus test system, considering a 4% average variation over the crisp power injections, i.e., a scenario comparable to our $\alpha = 1$. From their report, we can compute the average variation obtained for the voltage magnitudes, which is 0.92%. The average variation obtained with our method for $\alpha = 1$ is 1.22%. Thus, we can assume that both methods are very similar in terms of accuracy, since the size of the network must have an impact over the interval results size. However as stated in [9], another drawback of method B is that it does not converge when the injected power variation is set to 50%.

B. Complexity

Method A requires one matrix-vector multiplication after the computation of the crisp solution to propose a fuzzy solution. The matrix of size n used is the Jacobian of the Newton-Raphson power flow. Therefore method A requires $2n^2$ operations.

Method B is based on the interval Newton's method [13] and requires solving an interval linear system of the size of the Jacobian matrix at each iteration. By using the technique proposed by Rump in [17], this can be done in $5n^3$ operations.

The proposed method computes at each iteration the variable increment by the function Δ defined in Section II. This corresponds to solving a crisp system of linear equations and a linear program. The linear system can be solved in $\mathcal{O}(n^3)$ operations with a LU factorization and the linear program can be solved also in $\mathcal{O}(n^3)$ with the Simplex method [18]. Hence, the proposed method is expected to be slower than method A and comparable to method B in terms of performance for large systems.

V. CONCLUSIONS

Power flow simulation considering uncertainty is relevant to system operation and planning in today's complex, evolving energy context. This article proposes to address uncertainty with an original solution based on symetric fuzzy alegra. This method is compared with other related approaches from the literature, and proves to yield relatively accurate results at an acceptable algorithmic cost. Future works will consider a more comprehensive characterization of the solution and a measure of its optimality; simulations over larger networks considering various levels of uncertainty; as well as further comparison with other methods and Monte Carlo's simulations.

REFERENCES

[1] A. Vaccaro, C. Cañizares, and D. Villacci, "An affine arithmetic-based methodology for reliable power flow analysis in the presence of data uncertainty," *IEEE Trans. on Power Systems*, vol. 25, no. 2, pp. 624–632, 2010.

[2] M. Cortés-Carmona, R. Palma-Behnke, and G. Jiménez-Estévez, "Fuzzy arithmetic for the dc load flow," *IEEE Trans. on Power Systems*, vol. 25, no. 1, pp. 206–214, 2010.

[3] J. J. Grainger and W. D. Stevenson, "Power system analysis," 1994.

[4] Chun-Lien Su, "Probabilistic load-flow computation using point estimate method," *IEEE Trans. on Power Systems*, vol. 20, no. 4, pp. 1843–1851, 2005.

[5] R. N. Allan and M. R. G. Al-Shakarchi, "Probabilistic ac load flow," *Proceedings of the Institution of Electrical Engineers*, vol. 123, no. 6, pp. 531–536, 1976.

[6] A. M. Leite da Silva, V. L. Arienti, and R. N. Allan, "Probabilistic load flow considering dependence between input nodal powers," *IEEE Trans. on Power Apparatus and Systems*, vol. 103, no. 6, pp. 1524–1530, 1984.

[7] A. Dimitrovski and K. Tomsovic, "Boundary load flow solutions," *IEEE Trans. on Power Systems*, vol. 19, no. 1, pp. 348–355, 2004.

[8] A. Schellenberg, W. Rosehart, and J. Aguado, "Introduction to cumulant-based probabilistic optimal power flow (p-opf)," *IEEE Trans. on Power Systems*, vol. 20, no. 2, pp. 1184–1186, 2005.

[9] Z. Wang and F. L. Alvarado, "Interval arithmetic in power flow analysis," *IEEE Trans. on Power Systems*, vol. 7, no. 3, pp. 1341–1349, 1992.

[10] V. Miranda, M. A. A. C. Matos, and J. T. Saraiva, "Fuzzy load flow-new algorithms incorporating uncertain generation and load representation," in *Proceedings of the 10th PSCC*, 1990.

[11] E. M. Gouveia and M. A. Matos, "Symmetric ac fuzzy power flow model," *European Journal of Operational Research*, vol. 197, no. 3, pp. 1012–1018, 2009.

[12] D. Defour and M. Marin, "Fuzzygpu: a fuzzy arithmetic library for gpu," in *Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on*, 2014. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00856617

[13] R. E. Moore, *Methods and Applications of Interval Analysis*, ser. Studies in Applied and Numerical Mathematics. Society for Industrial and Applied Mathematics, 1979. [Online]. Available: http://books.google.ie/

[14] F. Milano, *Power System Modelling and Scripting*. London: Springer, 2010.

[15] J. Forrest, "The coin-or linear program solver (clp)," 2005.

[16] M. Heleno, J. Sumaili, J. Meirinhos, and M. A. da Rosa, "A linearized approach to the symmetric fuzzy power flow for the application to real systems," *International Journal of Electrical Power & Energy Systems*, vol. 54, no. 0, pp. 610 – 618, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142061513003384

[17] S. M. Rump, "On the solution of interval linear systems," *Computing*, vol. 47, no. 3-4, pp. 337–353, 1992.

[18] A. Schrijver, *Theory of linear and integer programming*. Wiley, 1998.