Analogy and Convergence of Levenberg's and Lyapunov-based Methods for Power Flow Analysis

Federico Milano, *Senior Member, IEEE*

***Abstract*— This letter focuses on the power flow problem and shows the formal analogy between the Levenberg's method and a fictitious ODE built using Lyapunov's second stability criterion. The letter also illustrates theoretical caveats and numerical issues of both methods. In particular, the case study, based on a 2383-bus system, shows that the fixed points of these methods are not necessarily a solution of the power flow problem.**

***Index Terms*— Levenberg's method, Levenberg-Marquardt's method, Lyapunov's function, Newton-Raphson's method, Power flow analysis.**

I. Introduction

The nonlinearity of power flow equations has been and is currently a challenge for system operators and practitioners. While several techniques have been proposed to handle ill-conditioned problems (see the literature review in [1]), convergence issues have not been fully solved so far. Due to the extreme importance for system operators of the availability of a solution of the power flow problem, this topic is evergreen in power system analysis.

The general formulation of the power flow problem is:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \,, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of unknowns and $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ are nonlinear, smooth equations defining the active and reactive power balance at network buses. Due to the nonlinearity of (1), it is possible to obtain a solution only through iterative methods (e.g., the well-known Newton-Raphson's method), which, in turn, obtain the solution of (1) through the computation of the fixed point of a given discrete map. So, even if the map converges, it is always necessary to verify that the fixed point of the map is actually also a solution of the original problem (1).

This letter focuses on the application to the power flow problem of two techniques, namely the Levenberg's method [2], [3] and the Lyapunov's function-based technique [4]. The contributions of the letter are as follows: (i) to prove that the Lyapunov-based method is a special case of the Levenberg's method; and (ii) to show that, while the Levenberg's and Lyapunov-based methods are claimed to be robust, they can be ill performing.

II. Levenberg's Method

The Levenberg's method is a nonlinear programming technique and as such has been successfully applied to optimization problems including optimal power flow analysis (see, for example, [5]). Let $\boldsymbol{x}_i$ be the value of the vector $\boldsymbol{x}$ at the i-th step of an iterative method applied to (1). The error vector associated with $\boldsymbol{x}_i$ is:

$$\boldsymbol{\epsilon}_i = \boldsymbol{f}(\boldsymbol{x}_i) \,. \tag{2}$$

A variation of $\boldsymbol{x}_i$, say $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \Delta\boldsymbol{x}_i$, will lead to the new error vector $\boldsymbol{\epsilon}_{i+1}$, which can be approximated through a first order Taylor's expansion:

$$\boldsymbol{\epsilon}_{i+1} = \boldsymbol{f}(\boldsymbol{x}_i + \Delta\boldsymbol{x}_i) \approx \boldsymbol{\epsilon}_i + \boldsymbol{J}_i \Delta\boldsymbol{x}_i \,, \tag{3}$$

where $\boldsymbol{J}_i = \nabla^T \boldsymbol{f}(\boldsymbol{x}_i)$ is the Jacobian matrix of $\boldsymbol{f}$ computed at $\boldsymbol{x}_i$. The Levenberg's method consists in determining the variation $\Delta\boldsymbol{x}_i$ that minimizes the sum of squares of the errors:

$$\eta_i \;=\; \boldsymbol{\epsilon}_{i+1}^T \boldsymbol{\epsilon}_{i+1} \;=\; \Delta\boldsymbol{x}_i^T \boldsymbol{J}_i^T \boldsymbol{J}_i \Delta\boldsymbol{x}_i + 2\Delta\boldsymbol{x}_i^T \boldsymbol{J}_i^T \boldsymbol{\epsilon}_i + \boldsymbol{\epsilon}_i^T \boldsymbol{\epsilon}_i \,. \tag{4}$$

The minimum of η_i is obtained at:

$$\nabla \eta_i(\Delta\boldsymbol{x}_i) = \boldsymbol{0} \,. \tag{5}$$

Hence, merging together (4) and (5), one has:

$$2\boldsymbol{J}_i^T \boldsymbol{J}_i \Delta\boldsymbol{x}_i + 2\boldsymbol{J}_i^T \boldsymbol{\epsilon}_i = \boldsymbol{0} \,. \tag{6}$$

Then (6) is solved for $\Delta\boldsymbol{x}_i$:

$$\Delta\boldsymbol{x}_i = -[\boldsymbol{J}_i^T \boldsymbol{J}_i]^{-1} \boldsymbol{J}_i^T \boldsymbol{\epsilon}_i \,, \tag{7}$$

and the vector of variables $\boldsymbol{x}$ is updated as follows:

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \Delta\boldsymbol{x}_i \,. \tag{8}$$

Observe that, from the property of the inverse of the product of two invertible square matrices, one has:

$$[\boldsymbol{J}_i^T \boldsymbol{J}_i]^{-1} \boldsymbol{J}_i^T = \boldsymbol{J}_i^{-1} [\boldsymbol{J}_i^T]^{-1} \boldsymbol{J}_i^T = \boldsymbol{J}_i^{-1} \boldsymbol{I} = \boldsymbol{J}_i^{-1} \,. \tag{9}$$

Hence, (7) is actually the increment as obtained by using the well-known Newton-Raphson's method. It is also well-known that the Newton-Raphson's method and, thus, (7), tends to diverge if the initial guess $\boldsymbol{x}_0$ is to far away from a fixed point of the map defined by (7)-(8). The main contribution of Levenberg has been to modify the objective function η_i in order to improve numerical convergence. This is achieved by including in the objective function the distance from the current point $\boldsymbol{x}_i$, as follows:

$$\eta_{\lambda,i} = \eta_i + \lambda \Delta\boldsymbol{x}_i^T \Delta\boldsymbol{x}_i \,, \tag{10}$$

where λ is called the *damping factor* of the Levenberg's method. Equation (10) leads to modify (7) as:

$$\Delta\boldsymbol{x}_i = -[\boldsymbol{J}_i^T \boldsymbol{J}_i + \lambda \boldsymbol{I}]^{-1} \boldsymbol{J}_i^T \boldsymbol{\epsilon}_i \,. \tag{11}$$

It is important to note that if λ is large, the effect of $\boldsymbol{J}_i^T \boldsymbol{J}_i$ vanishes. This fact will be discussed more in detail in Section III. Moreover, the value of λ has a strong impact on the number of iterations required to reach the convergence. This point is discussed in Section IV. This explains why, in most cases, a variant of (11) is preferred, as provided by Marquardt:

$$\Delta\boldsymbol{x}_i = -\left[\boldsymbol{J}_i^T \boldsymbol{J}_i + \lambda \,\mathbf{diag}\left(\boldsymbol{J}_i^T \boldsymbol{J}_i\right)\right]^{-1} \boldsymbol{J}_i^T \boldsymbol{\epsilon}_i \,, \tag{12}$$

where the identity matrix $\boldsymbol{I}$ is replaced with a diagonal matrix consisting of the diagonal elements of $\boldsymbol{J}_i^T \boldsymbol{J}_i$. This modification allows scaling the effect of the damping factor and reduces the number of iterations required to reach the convergence. Equation (12) is the well-known Levenberg-Marquardt's method. Several variants aimed to improve numerical stability and convergence have been proposed in the literature. See, for example, [6] and references therein.

III. Lyapunov-based Method

This subsection shows that the Lyapunov-based method proposed in [4] is actually a special case of (11). The starting point of the Lyapunov-based method is again the minimization of the sum of squares of $\boldsymbol{f}$ but, in this case, instead of using a discrete map, a continuous error function is defined:

$$\boldsymbol{\epsilon} = \boldsymbol{f}(\boldsymbol{x}) \,. \tag{13}$$

Then, a scalar function $V(\boldsymbol{x})$ is defined as follows:

$$V(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = \frac{1}{2}\eta \,. \tag{14}$$

Manuscript submitted to PES Letters, November 2014.

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

The main idea of this technique is to define a fictitious dynamic system for which $V(\boldsymbol{x})$ is a Lyapunov's function candidate. This can be obtained by defining the following set of first order differential equations:

$$\dot{\boldsymbol{x}} = -\boldsymbol{K}\frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} = -\boldsymbol{K}\left[\frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{x}}\right]^T \boldsymbol{\epsilon} = -\boldsymbol{K}\boldsymbol{J}^T\boldsymbol{\epsilon}\ , \qquad (15)$$

where $\boldsymbol{K}$ is a diagonal matrix of arbitrary positive coefficients. The interested reader can find in [7] the proof that $V(\boldsymbol{x})$ is a Lyapunov function of (15). This fact allows concluding that (15) is certainly stable and converges to an equilibrium point that corresponds to $V(\boldsymbol{x}) = 0$. Then any numerical integration scheme can be used to integrate (15), starting from the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$. In particular, the forward Euler's method gives:

$$\Delta\boldsymbol{x}_i = -h\boldsymbol{K}\boldsymbol{J}_i^T\boldsymbol{\epsilon}_i\ , \qquad (16)$$

where h is the time step of the Euler scheme. Note that (16) can be deduced from (11) by assuming that all elements k_{ii} of the diagonal matrix $\boldsymbol{K}$ are equal, i.e., $k_{ii} = \kappa,\ \forall i = 1,\dots,n$, and imposing $\lambda = 1/(h\kappa)$, with λ very large in order to nullify the effect of $\boldsymbol{J}_i^T\boldsymbol{J}_i$.

The equivalence stated above leads to the following remarks:

- The Lyapunov-based method has to be expected to converge with a relatively high number of iterations (at least with respect to the Newton-Raphson's method). This is confirmed by simulation results given in the next section.
- Given that (15) is an ODE and that (15) is basically a special case of the Levenberg's method, one could apply any sophisticated integration scheme to (11) or (12) to improve convergence, thus leading to a *continuous* Levenberg-Marquardt's method. This very idea but applied to the continuous Newton's method is exploited in [1].
- Observing (12) and (16), one can deduce:

  $$h\boldsymbol{K} = \left[\lambda\,\mathbf{diag}\left(\boldsymbol{J}_i^T\boldsymbol{J}_i\right)\right]^{-1}\ , \qquad (17)$$

  where the step length h functions as the inverse of the damping factor λ. The equality above can help define proper values of $\boldsymbol{K}$ and obtain convergence in case of stiff equations.

Another important remark concerns the ability of the map resulting from applying the Levenberg's method and its variants to converge to a solution of the original power flow problem (1). With this regard, let's define the function $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{J}^T\boldsymbol{\epsilon}$. The zeros of this function can be obtained by applying the Newton-Raphson's method, which leads to the following expression of the increment $\Delta\boldsymbol{x}_i$:

$$\Delta\boldsymbol{x}_i = -[\nabla^T\boldsymbol{\varphi}(\boldsymbol{x}_i)]^{-1}\boldsymbol{\varphi}(\boldsymbol{x}_i) = -[\boldsymbol{J}_i^T\boldsymbol{J}_i + (\nabla^T\boldsymbol{J}^T)\boldsymbol{\epsilon}_i]^{-1}\boldsymbol{J}_i^T\boldsymbol{\epsilon}_i\ , \qquad (18)$$

where $\nabla^T\boldsymbol{J}^T$ is a third-rank tensor that represents the gradient of $\boldsymbol{J}^T$. The interested reader can find further details on $\nabla^T\boldsymbol{J}^T$ in [1]. Based on (18), the following remarks are relevant:

- The map (7) is an approximation of (18). The approximation is "good" only if the term $(\nabla^T\boldsymbol{J}^T)\boldsymbol{\epsilon}_i \approx \mathbf{0}$, which certainly applies close to the solution. This fact explains why (7) tends to diverge if the initial guess $\boldsymbol{x}_0$ is far away from the solution.
- The Levenberg's method finds a solution of $\boldsymbol{\varphi}(\boldsymbol{x}) = \mathbf{0}$, not (1). Hence, the fixed points of (7) or its variants (11), (12) and (16) might not be a solution of (1). It is also straightforward to observe that a solution of (1) is certainly also a solution of (18), but the other way round does not apply, in general.

Another way to state the latter remark is that to *minimize* η_i (or its variants $\eta_{\lambda,i}$ and $\eta_{\Lambda,i}$) does not necessarily imply that, at the optimum $\boldsymbol{x}^*$, one has $\eta(\boldsymbol{x}^*) = 0$. Note also that, if $\eta(\boldsymbol{x}^*) \neq 0$, then the Lyapunov's conditions are not satisfied as $V(\boldsymbol{x}^*) \neq 0$ at the equilibrium point and, hence, (15) is not guaranteed to be stable or to converge to a solution of (1). Unfortunately, despite the noteworthy attempt to include slack variables proposed in [4], the issue regarding the potential mismatch of the solutions of (1) and (18) cannot be solved in practice, as shown in the next section.

TABLE I
PERFORMANCE OF DIFFERENT POWER FLOW SOLVERS

Method	λ	# of Iter.	time [s]	$\max\{\lvert\boldsymbol{\epsilon}\rvert\}$ [pu]
Newton-Raphson	0	4	0.022	$2.2 \cdot 10^{-9}$
LM, eq. (12)	1	3413	78.43	$4.8 \cdot 10^{-7}$
LM, eq. (12)	10	147	2.396	0.0475
LM, eq. (12)	100	85	1.394	0.1391
Lyapunov, eq. (16)	10^5	18	0.311	11.75
Method [6]	adaptive	14	0.286	1.2042

IV. Case Study

This case study reports a set of simulations based on the standard Newton-Raphson technique, the Levenberg-Marquardt's (LM) method, the improved LM method proposed in [6], as well as the Lyapunov-based method described in Section III. For the last method, the slack variables defined in [4] are not considered as these do not affect the convergence of the method itself. The network considered for this case study is a 2383-bus model of the Polish system, representing the winter 1999-2000 peak. The data of this system are available at http://www.pserc.cornell.edu/matpower/. All simulations are obtained using Dome [8] running on a 3.5 GHz Xeon. A flat start was used as initial condition and $\max\{|\Delta\boldsymbol{x}_i|\} \leq 10^{-3}$ was imposed as convergence criterion. Note that in most power system analysis tools, the convergence criterion considers $\max\{|\boldsymbol{\epsilon}|\}$, but this cannot be used in this case, as discussed below.

Results are shown in Table I. As expected, the Levenberg-based methods cannot guarantee that $\max\{|\boldsymbol{\epsilon}|\} \to 0$ in all cases. The higher the value of the damping factor λ, the smaller the number of iterations but, since the relative weight of $\boldsymbol{J}^T\boldsymbol{J}$ decreases, the convergence cannot be guaranteed. This is particularly evident for the Lyapunov-based method, for which the fixed point of the method does not correspond to a solution of the power flow problem. Intermediate results are obtained using a modified method such as the one proposed in [6] that defines an adaptive value of λ. Note, however, that the fixed point of this method depends also on the initial value of λ. For this case study, $\lambda_0 = 0.01$ has been used. This value has been heuristically determined.

V. Conclusions

This letter provides theoretical and practical insights on the Levenberg's method for power flow analysis. Based on the results of the case study, it can be concluded that this technique and its variants, do not, in general, provide a solution of the power flow problem, but rather of a different problem whose set of solutions contains those of the original power flow problem. Simulation results also show that the standard Newton-Raphson method outperforms the Levenberg's method and its variants in terms of both robustness and speed.

References

[1] F. Milano, "Continuous Newton's Method for Power Flow Analysis," *IEEE Trans. on Power Systems*, vol. 1, no. 24, pp. 50–57, Feb. 2009.

[2] P. J. Lagace, M. H. Vuong, and I. Kamwa, "Improving Power Flow Convergence by Newton Raphson with a Levenberg-Marquardt Method," in *IEEE PES General Meeting*, Pittsburgh, PA, July 2008, pp. 1–6.

[3] P. J. Lagace, "Power Flow Methods for Improving Convergence," in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, Oct 2012, pp. 1387–1392.

[4] Ning Xie, F. Torelli, E. Bompard, and A. Vaccaro, "Dynamic Computing Paradigm for Comprehensive Power Flow Analysis," *IET Generation, Transmission & Distribution*, vol. 7, no. 8, pp. 832–842, Aug. 2013.

[5] X. Li, Y. Li, and S. Zhang, "Analysis of Probabilistic Optimal Power Flow Taking Account of the Variation of Load Power," *IEEE Trans. on Power Systems*, vol. 23, no. 3, pp. 992–999, Aug. 2008.

[6] J. Fan, "The Modified Levenberg-Marquardt Method for Nonlinear Equations with Cubic Convergence," *Mathematics of Computation*, vol. 81, no. 277, pp. 447–466, Jan. 2012.

[7] F. Torelli, F. Milano, and A. de Bonis, "A General Power System Control Technique based on Lyapunov's Function," in *13th IEEE Int. Workshop on Signal Processing Advances in Wireless Communications*, Çeşme, Turkey, June 2012.

[8] F. Milano, “A Python-based Software Tool for Power System Analysis,” in *Procs. of the IEEE PES General Meeting*, Vancouver, BC, July 2013.