Implicit Continuous Newton Method for Power Flow Analysis

Federico Milano, *IEEE Fellow*

***Abstract*— This letter proposes an implicit form of the continuous Newton method to solve the power flow problem. The implicit formulation prevents the need to factorize the inverse of the Jacobian matrix of the power flow equations and allows exploiting implicit integration solvers. The backward Euler method is utilized in the letter for its L-stability and numerical robustness, which is independent from the step size. A 21,177-bus model of the ENTSO-E transmission system serves to show the performance of the proposed technique and to compare it with conventional methods considering both well-posed and ill-conditioned scenarios.**

***Index Terms*— Power flow analysis, nonlinear equations, implicit differential equations (IDEs), implicit integration schemes.**

I. Introduction

In their most general form, power flow equations are a set of nonlinear algebraic equations:

$$\mathbf{0} = \mathbf{g}(\mathbf{y}) \ , \tag{1}$$

where $\mathbf{y}$ ($\mathbf{y} \in \mathbb{R}^n$) and $\mathbf{g}$ ($\mathbf{g} : \mathbb{R}^n \mapsto \mathbb{R}^n$). The solution of (1) poses both theoretical and numerical challenges, which have been object of intense and continuous studies with the power system community since the first formulations in the mid 1950s [1].

The conventional Newton method to solve (1) leads to the following map [2]:

$$\mathbf{y}^{i+1} = \mathbf{y}^i - \mathbf{J}^{-1}(\mathbf{y}^i)\, \mathbf{g}(\mathbf{y}^i) \ , \tag{2}$$

where $\mathbf{J} = \mathbf{g}_y$ is the Jacobian matrix of the power flow equations and the super-indexes indicate the i-th iteration of the map. The initial value $\mathbf{y}^o$ is the initial guess. A fixed point of (2), namely $\mathbf{y}^{i+1} = \mathbf{y}^i$, is a solution of (1), which in practice is assumed to have been reached if $|\mathbf{y}^{i+1} - \mathbf{y}^i| < \epsilon$, where $\epsilon > 0$ is a given error tolerance.

II. Implicit Continuous Newton Method

Reference [3] proposes a formal analogy between (2) and a map representing a Forward Euler Method (FEM) for the following set of explicit nonlinear ordinary differential algebraic equations:

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}) = -\mathbf{J}^{-1}(\mathbf{y})\, \mathbf{g}(\mathbf{y}) \ , \tag{3}$$

and the i-th iteration of the FEM is:

$$\mathbf{y}^{i+1} = \mathbf{y}^i + h\, \mathbf{f}(\mathbf{y}^i) \ , \tag{4}$$

where h is the step size of the FEM, which in (2) is implicitly assumed to be $h = 1$. The convergence properties of (3) are thoroughly discussed in [3], which shows that sufficiently close to the equilibrium point, (3) is asymptotically stable.

In [4], the advantages of an implicit formulation of differential equations for transient stability analysis is discussed. Taking the cue from [4], (3) can be rewritten as a set of Implicit Differential Equations (IDEs):

$$\mathbf{J}(\mathbf{y})\, \dot{\mathbf{y}} = -\mathbf{g}(\mathbf{y}) \ , \tag{5}$$

which solves (1) in two cases: (i) equilibrium points for which $\dot{\mathbf{y}} = \mathbf{0}$; and (ii) singular points for which $\mathbf{J}(\mathbf{y})\, \dot{\mathbf{y}} = \mathbf{0}$ and $|\dot{\mathbf{y}}| \neq 0$. In the latter case, however, the system is not in stationary conditions.

Federico Milano is with School of Electrical & Electronic Engineering, University College Dublin, Dublin 4, Ireland. E-mails: federico.milano@ucd.ie

Federico Milano is supported by Science Foundation Ireland under the Investigator Programme award, project AMPSAS, Grant No. SFI/15/IA/3074.

Equations (5) can be conveniently integrated using an implicit integration scheme, whereas (3) can not due to the presence of the term $\mathbf{J}^{-1}$. The Backward Euler Method (BEM) appears a particularly promising scheme as it is L-stable [5]. This means that the BEM is intrinsically stable regardless the integration step size h, for $h > 0$. The i-th iteration of the BEM applied to (5) requires to solve the following set of nonlinear equations:

$$\mathbf{0} = \boldsymbol{\phi}(\mathbf{y}^i) = \mathbf{J}(\mathbf{y}^i) \left(\mathbf{y}^i - \mathbf{y}^{i-1}\right) + h\, \mathbf{g}(\mathbf{y}^i) \ , \tag{6}$$

where $\mathbf{y}^i$ is the vector of unknowns and $\mathbf{y}^{i-1}$ is the known vector of variables determined at the previous step. Equations (6) can be solved using a Newton iteration, as follows:

$$\mathbf{y}^i_{k+1} = \mathbf{y}^i_k - \boldsymbol{\phi}_y^{-1}(\mathbf{y}^i_k)\, \boldsymbol{\phi}(\mathbf{y}^i_k) \ , \tag{7}$$

where the sub-indexes indicate the k-th iteration of the Newton method that solves the i-th step of the integration of (5), and

$$\boldsymbol{\phi}(\mathbf{y}^i_k) = \mathbf{J}(\mathbf{y}^i_k) \left(\mathbf{y}^i_k - \mathbf{y}^{i-1}\right) + h\, \mathbf{g}(\mathbf{y}^i_k) \ . \tag{8}$$

The solution of (6), which in turn is the i-th step of the time domain integration of (5) is achieved when the map (8) returns $|\mathbf{y}^i_{k+1} - \mathbf{y}^i_k| < \delta$, where $\delta > 0$ is a given error tolerance. Then, the integration of (5) proceeds, until $|\mathbf{y}^{i+1} - \mathbf{y}^i| < \epsilon$, which also implies a stationary condition, i.e., $\dot{\mathbf{y}} \approx \mathbf{0}$.

The implicit BEM requires to iterate twice: in the inner loop over k for every time step i, and then in the main loop over i, to integrate the fictitious dynamic system (5). The solution of the inner Newton method for the i-th step requires to compute $\boldsymbol{\phi}_y(\mathbf{y}^i_k)$, whose expression can be deduced from (8):

$$\boldsymbol{\phi}_y(\mathbf{y}^i_k) = (1 + h)\, \mathbf{J}(\mathbf{y}^i_k) + \mathbf{H}(\mathbf{y}^i_k) \left(\mathbf{y}^i_k - \mathbf{y}^{i-1}\right) \ , \tag{9}$$

where $\mathbf{H} = \mathbf{g}_{yy}$ is the Hessian matrix of the power flow equations, namely a tensor of order 3.

The implementation of the implicit BEM based on (5) thus appears computationally demanding. However, this is not the case, for the following reasons.

- Since the BEM is L-stable, the step size h can be "large" without compromising the numerical stability of the method. A large h results in a small number of steps to reach the stationary condition $\dot{\mathbf{y}} \approx \mathbf{0}$ and thus $|\mathbf{y}^{i+1} - \mathbf{y}^i| < \epsilon$.
- The Newton method to solve the inner loop for the i-th step is particularly efficient and, in time domain analysis and for sufficiently small step size h, typically requires about two or three iterations to reach the condition $|\mathbf{y}^i_{k+1} - \mathbf{y}^i_k| < \delta$ [6].
- The calculation of the Hessian matrix $\mathbf{H}(\mathbf{y}^i_k)$ can be a deal-breaker for the efficiency and the implementation of the proposed method. As a matter of fact, the need to compute the Hessian matrix is one of the main reasons that have limited the utilization of robust methods such as [7] and [8]. Nevertheless, empirical tests have shown that the term $\mathbf{H}(\mathbf{y}^i_k) \left(\mathbf{y}^i_k - \mathbf{y}^{i-1}\right)$ is immaterial for the convergence of the inner loop.

The latter point has relevant consequences on the implementation and computational burden of the proposed implicit Newton method. Equation (9) can be conveniently approximated as:

$$\boldsymbol{\phi}_y(\mathbf{y}^i_k) \approx (1 + h)\, \mathbf{J}(\mathbf{y}^i_k) \ , \tag{10}$$

which leads to rewrite (7) with the following approximated BEM k-th iteration:

$$\begin{aligned}\mathbf{y}_{k+1}^{i} &\approx \mathbf{y}_{k}^{i} - (1+h)^{-1}\,\mathbf{J}^{-1}(\mathbf{y}_{k}^{i})\,\boldsymbol{\phi}(\mathbf{y}_{k}^{i}) \\ &= \mathbf{y}_{k}^{i} - (1+h)^{-1}\left[(\mathbf{y}_{k}^{i} - \mathbf{y}^{i-1}) + h\,\mathbf{J}^{-1}(\mathbf{y}_{k}^{i})\,\mathbf{g}(\mathbf{y}_{k}^{i})\right] \\ &= (1+h)^{-1}\{\mathbf{y}^{i-1} + h\,[\mathbf{y}_{k}^{i} + \mathbf{f}(\mathbf{y}_{k}^{i})]\}\ .\end{aligned} \tag{11}$$

There are at least two ways to further speed up the solution of the inner loop, as follows.

On one hand, one can approximate its solution, for example, by limiting the maximum number of k. Even if the inner-loop solution is not "exact," in fact, the main loop can eventually converge to the final equilibrium point. The case study considers the case of solving only one iteration of the inner loop, i.e., the inner-loop iterations are stopped at $k = 1$, regardless the convergence of the inner loop itself. Note that $\mathbf{y}_1^i = \mathbf{y}^{i-1}$ holds $\forall i$ due to the initialization of the inner loop, and, hence, $\mathbf{H}(\mathbf{y}_1^i)\left(\mathbf{y}_1^i - \mathbf{y}^{i-1}\right) = \mathbf{0}$. This implies that the Hessian matrix does not need to be calculated if approximating the inner-loop solution with its first iteration.

On the other hand, one can try to speed up the solution of the inner loop by taking advantage of the Dishonest Newton method that proved to be very efficient for time-domain integration [6]. Hence, (11) can be further simplified as follows:

$$\mathbf{y}_{k+1}^{i} \approx (1+h)^{-1}\{\mathbf{y}^{i-1} + h\,[\mathbf{y}_{k}^{i} + \mathbf{f}_o(\mathbf{y}_{k}^{i})]\}\ , \tag{12}$$

where

$$\mathbf{f}_o(\mathbf{y}_{k}^{i}) = -\mathbf{J}_o^{-1}\,\mathbf{g}(\mathbf{y}_{k}^{i})\ , \tag{13}$$

and $\mathbf{J}_o$ is a constant matrix, e.g., computed at the initial guess $\mathbf{y}^o$. Also this iteration setup is considered in the case study.

To complete this section, the proposed implicit continuous Newton method based on BEM integration is summarized in Algorithm 1 using pseudo code. In the pseudo code the function *stepSize* updates the step size h according to the number of iterations required by the inner loop to complete. An example of how to update h is discussed in the case study.

Algorithm 1 Main steps of the BEM

procedure IMPLICIT CONTINUOUS NEWTON METHOD
  Initial step size: $h \leftarrow 1$
  Set main-loop counter: $i \leftarrow 0$
  Initial variable guess: $\mathbf{y}^i \leftarrow \mathbf{y}^o$
  Main loop
  while $\max\{|\mathbf{g}(\mathbf{y}^i)|\} >$ *tolerance*$_1$ **do**
    Initial inner-loop counter: $k \leftarrow 0$
    Initial value: $\mathbf{y}_1^i \leftarrow \mathbf{y}^{i-1}$
    Inner loop
    for $k < k^{\max}$ **do**
      Solve (7) # *or approximated expression*
      if $\max\{|\mathbf{y}_{k+1}^i - \mathbf{y}_k^i|\} <$ *tolerance*$_2$ **then break**
      Update inner loop counter: $k \leftarrow k + 1$
    Adjust step size: $h \leftarrow$ *stepSize*(k)
    if *Inner Loop* converged **then**
      Update variables: $\mathbf{y}^i \leftarrow \mathbf{y}_{k+1}^i$
      Update main loop counter: $i \leftarrow i + 1$
      goto 7
    else
      if $i \geq i^{\max}$ **then break** # *divergence*
      goto 8
  return solution $\mathbf{y}^i$

III. CASE STUDY

In this case study, the properties and the performance of the proposed implicit continuous Newton method (ICNM) are compared through a static model of the ENTSO-E transmission system. The model includes 21,177 buses, 30,968 transmission lines and transformers, 1,144 zero impedance branches, 15,756 loads, and 4,828 generators. All simulations are obtained with Dome [9] running on a 3.5 GHz Intel Core i7 with 16 GB RAM.

Several other test systems, ranging from a few tens to a few thousands of buses have been tested with the proposed implicit continuous Newton method. Different networks require different numbers of iterations to reach convergence, but the behaviour and performance of the proposed implicit continuous Newton method as well as the main conclusions discussed below are the same. For this reason, only the largest network available to the author, i.e. the ENTSO-E system, is discussed below.

Four implementations of the ICNM are considered, namely: ICNM-$\mathbf{JH}$ that considers the expression with Jacobian and Hessian matrices (7); ICNM-$\mathbf{J}$ that utilizes the approximated expression (11) where the Hessian matrix is neglected; ICNM-$\mathbf{J}_1$ where the solution of the inner loop is approximated with its first iteration, i.e., $k^{\max} = 1$, and, hence, the Hessian matrix does not need to be calculated; and ICNM-$\mathbf{J}_o$ that utilizes (13) where the Jacobian matrix is kept constant. These three methods are compared with the standard Newton method (NM); a dishonest Newton method where the Jacobian matrix is factorized only once, i.e., at the first iteration (DNM); the optimal multiplier Newton method (OMNM) in polar coordinates defined in [8]; and the explicit continuous Newton method with 4^{th} order Runge-Kutta method (CNM-RK4) discussed in [3]. Note that, since the network includes non-conventional devices such as zero-impedance branches, which are efficiently modelled as variable power injections [10], the commonly-used fast decoupled power flow method (see, e.g., [11]) cannot be applied in this case as it requires that the only variables are bus voltage magnitudes and phase angles.

For the ICNMs, $\epsilon = \delta = 10^{-5}$ is used in all tests. The initial step size for the ICNM is $h = 1$. The step size is increased by 25% if the number of iterations of the inner loop to get to convergence is $k < 4$, and decreased by 25% if $k > 10$. The smaller h the more the iterations i of the main loop but the less the iterations of the inner loop. In time domain integration, the choice of the size of the step length h depends on the dynamics of the system. h is generally tuned in such a way to calculate enough points to properly plot the trajectories. This requires a relatively small h and, hence, the inner loop generally converges in no more than an handful of iterations. However, for the solution of the power flow, the main requirements is to reduce the total computing time to reach steady-state. The intermediate "transient" values taken by $\mathbf{y}$ during the iterations, in fact, are discarded. Finally, $h = 1$ is the value utilized in the conventional Newton method, which in this context can be interpreted as an explicit (forward) Euler method. Hence, using $h = 1$ allows for a fair comparison of the proposed method with existing ones.

Table I compares the statistics of the proposed method with conventional methods for the base-case flat-start power flow problem. Since the base case is feasible and well posed, all methods converge, being the conventional NM the least computationally demanding. Results also indicate that the statistics of the ICNM-$\mathbf{JH}$ and ICNM-$\mathbf{J}$ are effectively the same but for the CPU time, which is substantially lower for the ICNM-$\mathbf{J}$ as it does not require the calculation of the Hessian matrix $\mathbf{H}$. Thus, as anticipated in Section II, the term $\mathbf{H}(\mathbf{y}_k^i)\left(\mathbf{y}_k^i - \mathbf{y}^{i-1}\right)$ is not needed for the ICNM to converge. On

the other hand, approximating the Jacobian matrix with a constant matrix $\mathbf{J}_o$, while not preventing convergence, does not lead to a performance improvement. The best performing setup is ICNM-$\mathbf{J}_1$, i.e., the approach that uses $k^{\max} = 1$. This result suggests that the accuracy of the solution of the inner loop is not crucial for the convergence. In this scenario, the performance of ICNM-$\mathbf{J}$ and ICNM-$\mathbf{J}_1$ is comparable to that of a robust Newton method with adaptive step size, such as the OMNM. Note that, except for the ICNM-$\mathbf{J}_1$, the inner loop requires a relatively high number of iterations (at least compared to typical time domain analysis). This is due to the relatively high step length h. Reducing h can help reduce the number of inner-loop iterations but would increase the number of iterations of the main loop. By trial-and-error, $h = 1$ was found to be the initial guess that led to the minimum CPU times.

TABLE I
STATISTICS OF THE FLAT-START POWER FLOW ANALYSIS FOR THE ENTSO-E SYSTEM WITH VARIOUS SOLVERS

Method	Number of iterations		Number of	CPU time
	Main loop	Inner loop	factorizations	[s]
NM	7	–	7	0.195
DNM	28	–	1	0.183
OMNM	8	–	8	0.431
CNM-RK4	20	–	80	1.819
ICNM-$\mathbf{JH}$	5	21	21	0.992
ICNM-$\mathbf{J}$	5	21	21	0.543
ICNM-$\mathbf{J}_1$	11	1	11	0.338
ICNM-$\mathbf{J}_o$	8	118	1	1.832

The goal of the proposed ICNM is to solve ill-conditioned problems, i.e., power flow problems that have a solution but that cannot be solved with conventional methods and the standard *flat-start* initial guess, i.e., all bus voltage phase angles equal to zero. With this aim, the initial guess of the ENTSO-E system are modified with respect to the flat start to obtain a case for which conventional methods do not converge. Concretely, the initial-guess bus voltage phase angles are uniformly distributed in the interval $[-0.02, 0.02]$ rad, except for the slack bus, whose phase angle is zero. The same initial guess was used for all considered methods. This was obtained by using the same random number seed to generate the bus voltage phase angles for each solver tested in the case study.

Table II shows the results obtained with the same methods considered for the base case problem in Table I. Since generation and load profiles are unchanged, the sought solution is still the base-case one. However, due to the poor initial guess, NM, DNM and CNM-RK4 diverge. The OMNM, rather then diverging, finds in a local minimum for which $\mathbf{g}(\mathbf{y}) \neq \mathbf{0}$ and its optimal multiplier slowly goes to zero. On the other hand, all ICNM variants converge. Note that a variety of different symmetrical and asymmetrical intervals, not just $[-0.02, 0.02]$, were also tested, with similar results as those shown in Table II. The ICNM-$\mathbf{JH}$ and ICNM-$\mathbf{J}$ show same figures, thus confirming that the calculation of the Hessian matrix does not improve the robustness of the method. The best performing setup is again ICNM-$\mathbf{J}_1$ that is thus the iteration setup to show the most promising results. The ICNM-$\mathbf{J}_o$ converges with a slow pace but, since it requires only one factorization, its performance is similar to the ICNM-$\mathbf{JH}$. Overall, the best trade-off between performance and robustness is achieved with ICNM-$\mathbf{J}$.

TABLE II
STATISTICS OF THE ILL-CONDITIONED POWER FLOW ANALYSIS FOR THE ENTSO-E SYSTEM WITH VARIOUS SOLVERS

Method	Number of iterations		Number of	CPU time
	Main loop	Inner loop	factorizations	[s]
NM	Diverge	–	–	–
DNM	Diverge	–	–	–
OMNM	Local Min.	–	–	–
CNM-RK4	Diverge	–	–	–
ICNM-$\mathbf{JH}$	13	69	69	3.480
ICNM-$\mathbf{J}$	13	69	69	1.801
ICNM-$\mathbf{J}_1$	19	1	19	0.525
ICNM-$\mathbf{J}_o$	28	231	1	3.569

REFERENCES

[1] J. B. Ward and H. W. Hale, "Digital computer solution of power-flow problems," *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 75, no. 3, pp. 398–404, Jan. 1956.

[2] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-86, no. 11, pp. 1449–1460, Nov. 1967.

[3] F. Milano, "Continuous Newton's method for power flow analysis," *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 50–57, Feb. 2009.

[4] ——, "Semi-implicit formulation of differential-algebraic equations for transient stability analysis," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4534–4543, Nov. 2016.

[5] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*. New York: John Wiley & Sons, 2003.

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

[7] S. Iwamoto and Y. Tamura, "A load flow calculation method for ill-conditioned power systems," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 4, pp. 1736–1743, Apr. 1981.

[8] L. M. C. Braz, C. A. Castro, and C. A. F. Murati, "A critical evaluation of step size optimization based load flow methods," *IEEE Transactions on Power Systems*, vol. 15, no. 1, pp. 202–207, Feb. 2000.

[9] ——, "A Python-based software tool for power system analysis," in *Procs. of the IEEE PES General Meeting*, Vancouver, BC, July 2013.

[10] ——, "On the modelling of zero impedance branches for power flow analysis," *IEEE Trans. on Power Systems*, vol. 31, no. 4, pp. 3334–3335, July 2016.

[11] R. A. M. van Amerongen, "A general-purpose version of the fast decoupled load flow," *IEEE Transactions on Power Systems*, vol. 4, no. 2, pp. 760–770, May 1989.